

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky

DIPLOMOVÁ PRÁCE

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství

**Návrh a realizace panelu pro zobrazení
světového času**

Design and Realization of a World Time Panel

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství

Zadání diplomové práce

Student: **Bc. Tomáš Lipka**
Studijní program: N2649 Elektrotechnika
Studijní obor: 2601T004 Měřicí a řídicí technika
Téma: **Návrh a realizace panelu pro zobrazení světového času**
Design and Realization of a World Time Panel

Zásady pro vypracování:

Práce se bude zabývat zobrazením přesného času na závěsném panelu. Úkolem studenta je získat přesný čas z časového serveru a zobrazit jej na závěsném panelu tvořeném LED panely dle dodaného mechanického řešení.

1. Výběr vhodné technologie pro zobrazení data, času a místa.
2. Přesný čas a možnosti jeho získání (time servery).
3. Modulární vývojová platforma Tower (K60N512 nebo PXN20).
4. Návrh řídicí aplikace BareMetal nebo s pomocí RTOS MQX.
5. Využití vhodné vnitřní sběrnice pro připojení jednotlivých panelů.
6. Realizace řešení.
7. Testování řešení.
8. Dokumentace vytvořeného řešení.
9. Zhodnocení dosažených výsledků.

Seznam doporučené odborné literatury:

1. FREESCALE SEMICONDUCTOR. <i>Freescale MQX™ RTOS: Reference Manual</i>. 06/2012.
2. Freescale MQX™ Real-Time Operating System (RTOS). [online]. [cit. 2012-11-01]. Dostupné z: <http://www.freescale.com/webapp/sps/site/overview.jsp?code=MQXRTOS>
3. Freescale TWR-PXN20 [online]. c2012 [cit. 2012-11-01]. Dostupný z WWW: <http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=TWR-PXN20>.
4. HORSKÝ, Petr. Formica 4.2 : návrhový systém pro plošné spoje : editor plošného spoje a autorouter: příručka uživatele . 1. Praha : BEN, 1998. 190 s. ISBN 80-86056-34-1.
5. KŘIVKA, Ivo. Formica 4.2 : návrhový systém pro plošné spoje : schematický editor : příručka uživatele. 1. Praha : BEN, 1998. 248 s. ISBN 80-86056-35-X.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Jiří Kotzian, Ph.D.**

Datum zadání: 16.11.2012

Datum odevzdání: 07.05.2013



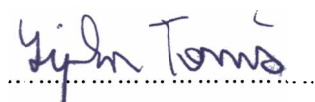
doc. Ing. Jiří Koziolek, Ph.D.
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

A handwritten signature in blue ink, reading "Lipka Tomáš", written over a horizontal dotted line.

V Ostravě dne 7. 5. 2013

Tomáš Lipka

Poděkování

Tímto způsobem bych chtěl poděkovat vedoucímu mé diplomové práce panu Ing. Jiřímu Kotzianovi, Ph.D. za cenné rady, dodání potřebných materiálů a konzultace spojené s vypracováním mého úkolu.

Zároveň bych velmi rád poděkoval panu Romanu Kopciovi za výrobu jednotlivých částí konstrukce podle výkresové dokumentace a panu Jakubu Lipkovi za pomoc s výkresovou dokumentací.

Abstrakt

Tato diplomová práce se zabývá návrhem a realizací závěsného panelu pro zobrazení světového času (TimePanel). TimePanel je sestaven z LED panelů a slouží k zobrazení přesného světového času vybraných lokalit s rozdílnými časovými pásmy. Přesný čas je získáván z časových serverů pomocí internetového (Ethernet) připojení. Výsledkem celé diplomové práce je funkční a přehledný TimePanel s dobrou viditelností zobrazující světový čas, datum a název lokality.

Klíčová slova

Formica, PCB, SNTP ,I2C , RTOS ,MQX , Power PC, PXN20 ,TimePanel, Expandér

Abstract

This Master's thesis deals with design and realisation of the aerial panel, which displays world time (TimePanel). TimePanel is constructed of LED panels and is instrumental towards displaying exact world time of selected locations with diverse time zones. Exact time is obtained from timeservers by means of the Internet (Ethernet) connection. The result of this thesis is functional and synoptical TimePanel with good visibility, which displays the world time, date and the name of the locality.

Key Words

Formica, PCB, SNTP ,I2C , RTOS ,MQX , Power PC, PXN20 ,TimePanel , Expander

Seznam použitých symbolů a zkratk

DPS	– Deska plošných spojů
FIFO	– (First In, First Out), speciální paměť
GPIO	– General Purpose Input/Output
I2C	– Inter-Integrated Circuit
LED	– Light-emitting diode
RS232	– Standardní pro sériové rozhraní
Ethernet	– Komunikační sběrnice
RTCS	– Real-Time TCP/IP Communication Suite
SMD	– Surface Mount Devices
SMT	– Surface mount technology
UART	– Universal Asynchronous Receiver Transmitter
USB	– Universal Serial Bus
RTOS	– Real-Time Operating System
IP adresa	– Internet Protocol Address - jedinečná internetová adresa
TimePanel	– Zařízení pro zobrazení světového času (lokalita, datum a čas)
LED	– Light Emitting Diode
TWI	– Two Wire Interface
JTAG	– Joint Test Action Group
NEXUS	– Debugging interface for embedded systems
IP	– Internet Protocol
UML	– Unified Modeling Language
DHCP	– Dynamic Host Configuration Protocol
MQX	– Operační systém reálného času od FreeScale
SHELL	– Příkazový procesor (řádek)
C	– Programovací jazyk nízké úrovně
SELČ	– Středoevropský letní čas
SEČ	– Středoevropský čas
GMT	– Čas na nultém poledníku -zeměpisná šířka (Greenwich Mean Time)
UTC	– Světový čas, stejný pro celý svět (Universal Coordinated Time)
SNTP	– Simple Network Time Protocol
DST	– Daylight Saving Time
TELNET	– Telecommunications Network (port 23)

Obsah

Obsah	- 1 -
1 Úvod	- 3 -
2 Rozbor zadání	- 4 -
2.1 TimePanel	- 4 -
2.2 Seznam požadavků.....	- 5 -
2.3 Stávající návrhy řešení.....	- 5 -
3 Použité softwarové a hardwarové nástroje	- 6 -
3.1 Operační systémy	- 6 -
3.1.1 Operační systémy reálného času (RTOS).....	- 6 -
3.1.2 Freescale MQX RTOS.....	- 7 -
3.1.3 Vývojové prostředí	- 9 -
3.2 Zobrazovací jednotky.....	- 10 -
3.2.1 LED diody	- 10 -
3.2.2 LCD displeje.....	- 11 -
3.2.3 Fluorescentní displeje	- 12 -
3.2.4 Elektromechanické zobrazovací prvky.....	- 12 -
3.2.5 TFT obrazovky	- 12 -
3.2.6 Možnosti zapojení segmentových a maticových LED displejů.....	- 12 -
3.3 Komunikace mezi moduly v rámci zařízení (I2C, SPI).....	- 13 -
3.3.1 Sběrnice SPI	- 13 -
3.3.2 Mezi desková komunikace - Sběrnice I2C	- 14 -
3.4 Synchronizace času.....	- 16 -
3.4.1 Synchronizace s Protokoly NTP a SNTP	- 16 -
3.4.2 Letní čas – Local Time	- 18 -
4 Realizace navrženého řešení.....	- 20 -
4.1 Mechanická část – konstrukce	- 20 -
4.1.1 Konstrukční návrhy TimePanelu	- 20 -
4.1.2 Uspořádání LED modulů v konstrukci	- 22 -
4.1.3 Konečné řešení konstrukce TimePanelu.....	- 22 -
4.2 Návrh a realizace schémat a DPS	- 25 -
4.2.1 LED panely – moduly Time panel 1 až 3	- 25 -

4.2.2	Propojení modulů TimePanelu	- 27 -
4.2.3	Společné součástky pro moduly TimePanelu	- 28 -
4.2.4	Modul TimePanel_1	- 31 -
4.2.5	Modul TimePanel_2	- 33 -
4.2.6	Modul TimePanel_3	- 35 -
4.2.7	Rozšiřující deska „Expansion board“	- 38 -
4.3	Výběr hardwarových součástí TimePanelu	- 40 -
4.3.1	Řídící jednotka – Freescale Power PC - modul PXN20	- 40 -
4.3.2	Napájecí zdroje	- 43 -
4.4	Softwarová - realizační část	- 44 -
4.4.1	Příkazy a nastavení periférii v MQX	- 44 -
4.4.2	Přepínání lokalit TimePanelu	- 45 -
4.4.3	Aplikace pro TimePanel	- 48 -
5	Testování a oživování HW a SW	- 71 -
5.1	Předrealizační testování HW a SW	- 71 -
5.2	Rozšiřující deska „Expansion_BOARD“	- 73 -
5.3	Moduly TimePanelu	- 74 -
5.4	Kompletní test TimePanelu	- 75 -
6	Závěr	- 76 -
7	Použitá literatura	- 78 -
	Seznam příloh	- 80 -

1 Úvod

V reálném životě je spousta odlišných pracovních oborů s různými zaměřenými. V některých těchto pracovních oborech může být například zapotřebí nepřetržité nebo občasné sledování přesného světového času a data ve vybraných lokalitách s odlišnými časovými pásmy. K tomu by mohlo sloužit zařízení, umístěné na přehledném a dobře viditelném místě ve vnitřních prostorech. Takové zařízení by bylo kupříkladu vhodné v mezinárodní firmě Freescale, kde je nutné zohlednit čas v dalších pobočkách firmy či čas u zákazníků. Pohledem na zařízení HW či SW vývojář zjistí přesný světový čas vybrané lokality.

Hlavní úlohou TimePanelu je tedy zobrazení přesného světového času vybraných šesti lokalit, ve vnitřních prostorech. TimePanel by měl zobrazovat přesný světový čas a to nejlépe získaný s *Time Serveru* pomocí internetového připojení. Důležité je také aby TimePanel zavěšený o strop, nejlépe uprostřed místnosti byl dobře přehledný a viditelný z obou stran.

Diplomová práce se zabývá návrhem a realizací zařízení pro zobrazení světového času a data v různých lokalitách (TimePanel). Řešení této práce spočívá v návrhu a realizaci jednotlivých displejů pro zobrazení času, data a lokality. Dále pak volbě vhodné sběrnice a řídicí jednotky pro komunikaci s jednotlivými displeji, vytvoření softwarové aplikace pro TimePanel a na závěr je zaměřena na návrh, realizaci konstrukce a sestavení celého TimePanelu.

Současná diplomová práce částečně vychází ze stávajících dílčích navrhovaných projektů studentů (HW, SW, konstrukce...). Před tím než bude započatá práce na návrhu a realizaci TimePanelu, je nutné se seznámit s tím co vůbec TimePanel je, k tomu slouží kapitola 2. V ní je popsáno využití TimePanelu, z čeho se skládá a jaké požadavky by měl splňovat.

V kapitole 3. jsou popsány vlastnosti, výhody i nevýhody použitých softwarových a hardwarových nástrojů v TimePanelu. Popisuje použitý operační systém s vývojovým prostředím a možnosti a způsoby synchronizace času. Dále zde jsou uvedeny různé typy zobrazovacích jednotek, a vhodné sběrnice pro mezi modulovou komunikaci v TimePanelu.

Kapitola 4. se zabývá průběhem realizace jednotlivých částí navrženého řešení TimePanelu. První, mechanická část, je věnována návrhu a realizaci konstrukce. Druhá část je zaměřena na návrh a realizaci DPS modulu TimePanelu a také na výběr hardwarových součástí. A v třetí, softwarové části je popsána aplikace pro TimePanel s použitím UML diagramu.

Průběh testování hardwarů a softwarů na nepájivém poli, které probíhalo před započítím realizační části řešení je uveden v kapitole 5. Dále jsou zde popsány jednotlivé postupy při oživení a testování rozšiřující desky a modulu TimePanelu. V závěru kapitoly je zaznamenán průběh kompletního softwarového a hardwarového testování TimePanelu.

2 Rozbor zadání

Před tím než bude započatá práce na návrhu a realizaci TimePanelu, je nutné se seznámit s tím, co TimePanel je, k čemu slouží, z čeho se skládá a jaké požadavky by měl splňovat.

2.1 TimePanel

TimePanel je zařízení pro zobrazení světového času ve vnitřních prostorách. Tyto hodiny budou řízeny mikrokontrolérem Power PC - PXN20 od firmy Freescale, a budou zobrazovat čas v jednotlivých pobočkách firmy Freescale. Tento světový čas v lokalitách se bude přepínat buď automaticky s určitým časovým intervalem anebo manuálně pomocí tlačítka. Na TimePanelu bude kromě času zobrazené i datum a název lokality. V rámci studentských projektů byly vytvořeny předběžné návrhy konstrukce TimePanelu. Konečný návrh řešení vychází z těchto podkladů a tyto návrhy vylepšuje a rozšiřuje. (Obr. 1).

V TimePanelu budou použity tři různé typy modulů:

- **Velký:** zobrazení hodin – **TimePanel 2**
- **Menší:** zobrazení data – **TimePanel 3**
- **Maticový:** zobrazení názvu města – **TimePanel 1**



Obr. 1 TimePanel - 3D model s AutoDesk Inventor

Tyto LED moduly budou s mikroprocesorem komunikovat pomocí sběrnic I2C. Z toho plyne, že každý panel by měl být autonomní. Přesným zdrojem času na TimePanelu bude SNTP server připojený k zařízení přes Ethernet, který bude v pravidelných intervalech aktualizovat čas. V případě že SNTP server nebo připojení k internetu nebude dostupné, je možné čas nastavit manuálně přes terminál.

Nastavit čas a datum lze tedy:

- Stažením času z internetu (SNTP Client – Time Servery)
- Ručním nastavením (příkazový řádek SHELL – TELNET)

Na TimePanelu jsou zobrazeny časy, data a zkratky názvů vybraných světových lokalit (Tabulka 1). V této tabulce jsou také uvedeny časové pásma - posuny UTC těchto lokalit.

Zobrazované lokality na TimePanelu			
Zkratka	Město	Země	Časový posun (UTC)
ROZ-CZ	Rožnov	Česká republika	+1 hodina
AUS-TX	Austin	Texas	-6 hodin
GDL-MX	Guadalajara	Mexiko	-6 hodin
EKB-UK	East KilBride	Velká Británie	±0 hodin
NOI-IN	Noida	Indie	+5.30 hodin
SHA-CH	Shanghai	Čína	+8 hodin
BJS-CH	Beijing	Čína	+8 hodin

Tabulka 1 Zobrazované lokality na TimePanelu

Velkou výhodou tohoto zařízení je, že se jedná o modulární systém složený z 22 modulu (tři různých typů) a v případě jakékoliv poruchy nebo závady na modulu (displeji), stačí jen poškozený modul vyměnit za funkční a nastavit mu stejnou adresu (DIP přepínač) jako u poškozeného modulu. Není zapotřebí provádět demontáž a výměnu celé elektroniky nebo přehrání firmware.

2.2 Seznam požadavků

- Umístěny na chodbě pod stropem.
- Oboustranný panel.
- Zdroj času – *Time server – Ethernet*.
- Přehledný panel s dobrou viditelností.
- Velikost hodin cca 10-15 cm.
- Vteřiny nejsou zobrazeny.
- Napájený z 230V.
- Zobrazení přesného času (včetně data) pro různá místa na světě.
- Tlačítko pro přepínání lokalit a volbu režimu přepínání (manuální a automatické).
- Nastavení z konzoly / web serveru.
- Použití Power PC – PXN20 nebo K60 modulu ze systému Freescale Tower System.
- Možnost řetěžit panely.
- ...

2.3 Stávající návrhy řešení

Současná diplomová práce částečně vychází ze stávajících dílčích navrhovaných projektů studentů. Tyto dílčí návrhy se týkaly návrhu schémat LED panelu, výběru součástek, řídicí jednotky, návrh UML diagramu pro TimePanel a především návrhem konstrukčních řešení TimePanelu.

3 Použité softwarové a hardwarové nástroje

V této kapitole jsou shrnuty vlastnosti, parametry, výhody i nevýhody použitých softwarových a hardwarových nástrojů v TimePanelu. Popisuje použitý operační systém s vývojovým prostředím a možnostmi a způsoby synchronizace času. Dále jsou v kapitole popsány různé typy zobrazovacích jednotek, vhodné sběrnice pro mezi modulovou komunikaci a konstrukční návrhy řešení TimePanelu. V závěru kapitoly je stručný popis použitého hardwaru jako napájecího zdroje a řídicí jednotky.

3.1 Operační systémy

Operační systém je základní programové vybavení počítače, které tvoří rozhraní mezi uživatelem. Skládá se z jádra a systémových nástrojů. S jejich pomocí umožňuje uživatelům jeho ovládání. Poskytuje aplikační rozhraní (*API*) pro využívání služeb operačního systému uživatelským programovým vybavením (software, aplikace). Operační systém rovněž přiděluje systémové zdroje spuštěným aplikacím. Způsob jak jsou zdroje přidělovány spolu s rychlostí a systémovou režii přidělování zdrojů je pro vestavěné řídicí systémy klíčová. Operační systém je první software, který je zaveden zavaděčem po zapnutí zařízení a tento software není nikdy ukončen na rozdíl od aplikací, které mohou běžet po omezenou dobu.

Operační systémy můžeme pro zjednodušení rozdělit na dvě základní skupiny. První skupinou jsou operační systémy v nasazení, kde je vidíme nejčastěji a to v pracovních stanicích a osobních počítačích pro kancelářskou, úřednickou nebo vývojovou práci (*Windows 7*). Druhá skupina operačních systémů jsou operační systémy pro vestavěné řídicí systémy. S operačními systémy z této skupiny se setkáváme nejčastěji, ale pro charakter zařízení jsou skryté a většina uživatelů nemá o jejich existenci ani tušení (*RTOS MQX*). [4] [13]

3.1.1 Operační systémy reálného času (RTOS)

Operační systémy pro práci v reálném čase neposkytují komfortní uživatelské rozhraní. Většinou ani nemají připojen displej nebo klávesnici jako nonRT operační systémy. Jejich vlastnosti však umožňují nasazení do aplikací, kde čas je klíčovou částí dané aplikace. Pokud selžou, hrozí nebezpečí poškození zdraví nebo majetku. Uživatele nasazený operační systém *„nezajímá“*, přesto systém musí pracovat dny či roky bez vypnutí služby. [4] [13]

RTOS lze rozdělit podle náročnosti na plnění časových limitů do tří skupin následovně:

- **Tvrdé časy (*hard RTs*)** – požadovaná výkonnost musí být vždy dodržena. Nedodržení času způsobí selhání systému. Pozdní data = špatné data. Příklady: palubní počítač, jednotka ABS, řídicí systém robota, medicínské přístroje.
- **Měkké časy (*soft RTs*)** – Řídíme se průměrnými časy – celková průchodnost systému. (Příležitostné nedodržení, Nedodržení s malými odchylkami, Příležitostné zrušení) Pozdní data = mohou být dobrá data. Příklady: letecký rezervační systém, online databáze, myčka, mikrovlnná trouba, automobilová navigace.
- **Tuhé časy (*firm deadlines*)** – míra roste s časem nedodržení (zpočátku měkké postupně hard). Příklady: plicní ventilátor, multimediální systémy v automobilu.

Dále je možné OS/RTOS rozdělit na dvě základní kategorie podle zavedených úloh:

- **Kompilované operační systémy** (*FreeRTOS, MQX*)
- **Samostatné operační systémy** (*QNX a LinuxLink*)

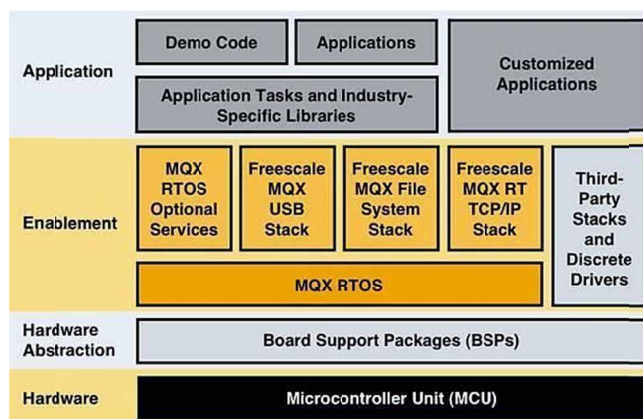
Chování operačního systému (RTOS) je závislé především na jeho základní struktuře. Určující je jak jsou vzájemně propojeny jednotlivé jeho části – funkce a jaké jsou jejich závislosti. Struktura RTOS udává také mimo bezpečnosti systému také jeho zabezpečení.[4][13]

Operační systémy můžeme dle struktury rozdělit takto:

- **Konvenční** (*FreeRTOS*)
- **Monolitické** (*MQX*)
- **Mikrojádru** (*QNX Neutrino*)

3.1.2 Freescale MQX RTOS

Real-time operační systém MQX není OS v podobě, jak jej známe dnes například v podobě OS Windows. MQX je prakticky run-time knihovna funkcí, které může programátorem psaná aplikace využívat, aby mu jednak ulehčila práci (snadná práce se vstupy/výstupy, obsluhy přerušení, datová komunikace, správa paměti apod.) a také aby se jeho program mohl stát v reálném čase běžící multitasking aplikací. Tedy programem, kde *paralelně* vykonává více funkcí (částí programu) současně a dělí se o pracovní čas procesoru (Obr. 2).



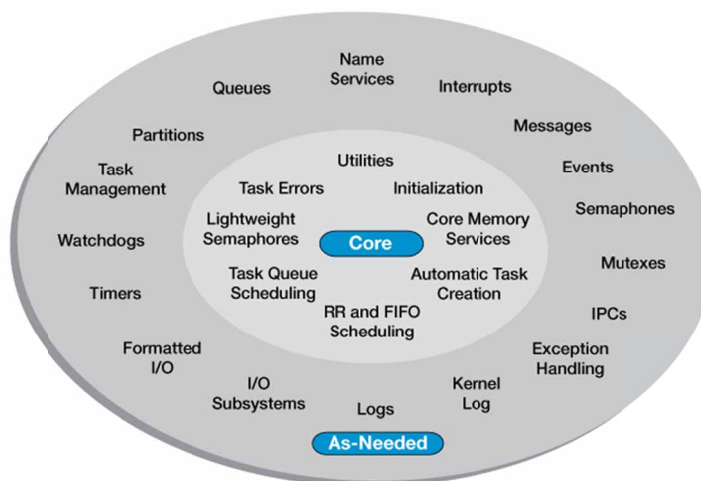
Obr. 2 Architektura MQX RTOS[11]

Díky již vytvořeným propracovaným MQX funkcím se tak programátor nemusí zabývat nastavováním bitů na úrovni jednotlivých registrů, řízení běhu programu, ošetřování hardwarových přerušení, ale prostě již využívá vhodného nastavení a volání různých MQX funkcí (tzv. komponent), které již sami v reálném čase zajistí a monitorují vše potřebné. Programátor aplikace se tak nemusí detailně zabývat samotnou hardwarovou strukturou procesoru, pro který program vytváří, ale již pracuje na vyšší úrovni a může plně využívat možností programovacího jazyku C.[16] [11]

3.1.2.1 Komponenty a příklady použití MQX

MQX se skládá z komponent jádra a doplňkových rozšiřujících komponent. V případě komponent jádra se pak pouze ty, které jsou využívány MQX nebo volány aplikací, se vloží při překladač do

programu. Tím se minimalizuje velikost kódu. Ke splnění dalších požadavků, které rozšiřují aplikace a konfigurují komponenty jádra, se pak dále vkládají další doplňkové komponenty (Obr. 3).[11] [16]



Obr. 3 Komponenty MQX RTOS [16]

3.1.2.2 Inicializace MQX

Dá se říct, že mezi hlavní a nejdůležitější části jádra patří komponenta ,Inicializace'. Tato funkce totiž inicializuje hardware a spustí MQX. To následně vytvoří a spustí úlohy (*tasks*), které jsou v inicializaci definovány jako ,autostart'. Bez komponenty ,Inicializace' by tedy se MQX ani program nespustil a neběžel.

Seznam úloh (*task template list*) definuje takovou zahajovací skupinu, ze kterých mohou být v procesoru úlohy vytvořeny. Po inicializaci MQX vytvoří jeden exemplář každé úlohy, která na seznamu figuruje s položkou ,autostart task'. Tyto úlohy pak mohou dále vytvářet a spouštět další úlohy. Konec seznamu úloh je vždy zakončen ,nulovou úlohou'. [11] [16]

```
TASK_TEMPLATE_STRUCT MQX_template_list[] =
{
    /* Task Index, Function, Stack, Priority, Name, Attributes, Param, Time Slice */
    { MAIN_TASK,      main_task,      1500,    9,    "main",      MQX_AUTO_START_TASK, 0, 0 },
    { IIC_1_TASK,     iic_1_task,     2000,    6,    "iic_1",     0, 0, 0 },
    { LOKALITA_TASK,  lokalita_task, 2000,    9,    "lokalita",  0, 0, 0 },
    { BTN_TASK,       btn_task,       2000,    9,    "btn",       0, 0, 0 },
    { TELNET_TASK,    telnet_task,    2000,    9,    "telnet",    0, 0, 0 },
    { SHELL_TASK,     shell_task,     2000,    5,    "shell",     0, 0, 0 },
    { IIC_2_TASK_1,   iic_2_task_1,   2000,    7,    "iic_2_1",   0, 0, 0 },
    { IIC_2_TASK_2,   iic_2_task_2,   2000,    7,    "iic_2_2",   0, 0, 0 },
    { IIC_2_TASK_3,   iic_2_task_3,   2000,    7,    "iic_2_3",   0, 0, 0 },
    { IIC_2_TASK_4,   iic_2_task_4,   2000,    7,    "iic_2_4",   0, 0, 0 },
    { IIC_2_TASK_5,   iic_2_task_5,   2000,    7,    "iic_2_5",   0, 0, 0 },
    { IIC_2_TASK_6,   iic_2_task_6,   2000,    7,    "iic_2_6",   0, 0, 0 },
    { POKUS,          pokus_task,      2000,    7,    "pokus",     0, 0, 0 },
    { 0,              0,              0, 0, 0, 0, 0, 0 }
};
```

Obr. 4 Ukázka seznamu úloh v MQX

3.1.2.3 Řízení a správa úloh

Další komponentou jádra je řízení úlohy (*Task management*). Automaticky spuštěné úlohy po spuštění MQX totiž mohou dále spravovat (vytvářet, spouštět, zastavovat, rušit) další úlohy běžícího programu. Lze provádět i několikanásobné spuštění stejné úlohy a není definován žádný

limit na počet běžících úloh v aplikaci. Tyto úlohy koexistují, protože mají unikátní identifikaci. MQX udržuje každou úlohu ukládáním jejího obsahu (program counter, registry a zásobník). Zároveň lze dynamicky za běhu provádět změnu atributů každé úlohy podle potřeby.

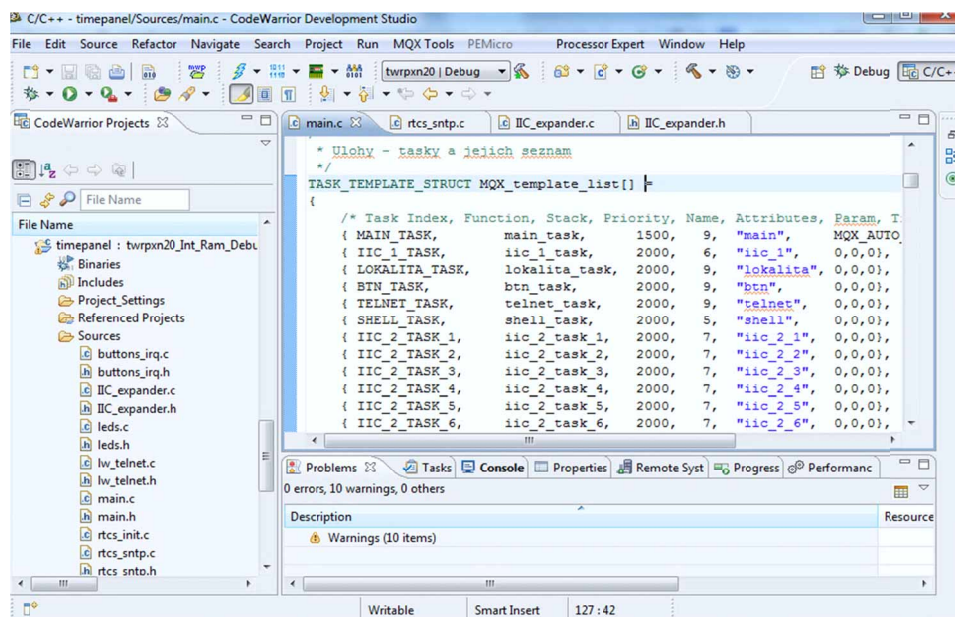
Po ukončení každé úlohy pak MQX automaticky uvolní paměť a její prostředky pro další úlohy. Pro každou úlohu lze také specifikovat výstupní funkci (*exit function*), kterou MQX zavolá při ukončování úlohy, nebo správce výjimek (*exception handler*), který MQX zavolá, pokud za běhu úloh dojde k nějaké neočekávanému stavu. [11] [16]

3.1.2.4 I/O ovladače

Mezi doplňkové komponenty pak patří I/O ovladače (*I/O drivers*) využívané na úrovni BSP. Skládají se z formátovaných I/O a I/O podsystémů. Pro formátované I/O poskytuje MQX knihovny funkcí, což jsou API pro I/O podsystémy. Ovladače I/O zařízení jsou dynamicky (nebo v reálném čase) instalované softwarovými balíčky, které poskytují přímý přístup k hardwaru systému, na kterém MQX a program aplikace běží. [11] [16]

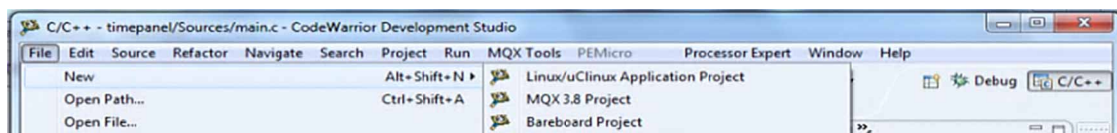
3.1.3 Vývojové prostředí

Pro vývoj aplikace pro TimePanel bylo vybráno vývojové prostředí CodeWarrior 10.2 s MQX 3.8. CodeWarrior je integrované vývojové prostředí (*IDE*) pro operační systémy Microsoft Windows, Linux a vestavěné systémy. Je vyvíjen společností Freescale Semiconductor. Soustřeďuje se především na nástroje pro jazyk C a C++, ale zahrnuje také Pascal a Java kompilátory. CodeWarrior 10.2 je postavený na Eclipse (Obr. 5).



Obr. 5 Vývojové prostředí CodeWarrior 10.2

Jelikož CodeWarrior 10.2 neobsahuje po nainstalování doplněk MQX, je zapotřebí jej doinstalovat (Obr. 6).



Obr. 6 CodeWarrior s MQX 3.8

3.2 Zobrazovací jednotky

Pro realizaci elektronického informačního panelu lze využít různé technologie zobrazovacích jednotek. Nejčastěji se však využívají LED diody, LED displeje, LCD displeje, elektromagnetické nebo fluorescentní prvky a v poslední době TFT obrazovky.

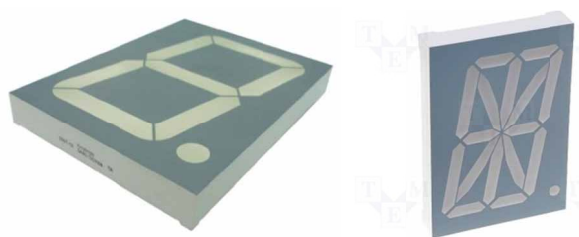
3.2.1 LED diody

Světlo-vyzařující dioda (LED) je elektronická polovodičová součástka obsahující přechod P-N. Prochází-li přechodem elektrický proud v propustném směru, přechod vyzařuje (emituje) nekoherentní světlo s úzkým barevným spektrem. Základní monokrystaly diod bývají překryty kulovými vrchlíky z epoxidové pryskyřice nebo akrylového polyesteru. Materiály, z nichž se LED vyrábějí, totiž mají poměrně vysoký index lomu a velká část vyzařovaného světla by se odrážela totálním odrazem zpět na rovinném rozhraní se vzduchem. [9]

Oproti jiným elektrickým zdrojům světla (žárovka, výbojka, doutnavka) mají LED tu výhodu, že pracují s poměrně malými hodnotami proudu a napětí. Z toho vyplývá jejich užití v displejích. Konstrukce displejů bývá zásadně dvojitá:

3.2.1.1 Segmentové – LED

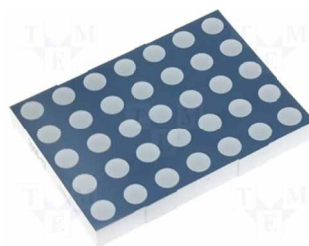
Segmentové – LED displeje (skupiny diod) jsou uspořádány do segmentů. Pro zobrazení číslic osmičkové, desítkové a šestnáctkové soustavy se používají sedmi segmentové displeje. Osmý segment tvoří desetinnou tečku. Pro zobrazení alfanumerických znaků (zpravidla bez diakritiky) se používají šestnácti segmentové displeje (Obr. 7). [9]



Obr. 7 7-segmentový displej SA40-19SRWA použitý v TimePanelu_2 a 16-segmentový displej

3.2.1.2 Maticové – LED

Maticové – LED displeje jsou uspořádány do matice a umožňují zobrazení grafické informace. Běžné uspořádání továrně vyráběných matic je 5x7, 8x8, 16x16 a 32x32 bodů. Pro zobrazení větší plochy se musí matice skládat do větších konstrukčních celků. Matici je možné sestavit také z diskretních LED diod. Ovládání takových zobrazovacích prvků bývá řešeno pomocí posuvných registrů připojených např. k rozhraní SPI. Velmi často se používá časový multiplex – dynamický režim řízení - přepínání sloupců (Obr. 8). [9]



Obr. 8 Maticový displej TA40-11SRWA použitý v TimePanelu_1

3.2.2 LCD displeje

Displej z tekutých krystalů (*Liquid crystal display - LCD*) je tenké a ploché zobrazovací zařízení skládající se z omezeného počtu barevných nebo monochromatických bodů (segmentů) seřazených před zdrojem světla nebo odrazovou plochou. Vyžaduje poměrně malé množství elektrické energie, proto se často používá v přístrojích s bateriovým napájením.

Každý pixel LCD se skládá z molekul tekutých krystalů uložených mezi dvěma průhlednými elektrodami a mezi dvěma polarizačními filtry, přičemž osy polarizace jsou na sebe kolmé. Bez krystalů mezi filtry by bylo světlo procházející jedním filtrem blokováno filtrem druhým. Molekuly tekutých krystalů jsou bez vnějšího elektrického pole neuspořádané. Elektrické pole způsobí, že se molekuly srovnají s mikroskopickými drážkami na elektrodách. Drážky na elektrodách jsou vzájemně kolmé, takže molekuly srovnány do spirálové struktury točí polarizaci procházejícího světla o 90 stupňů, což mu umožňuje projít i druhým filtrem. Polovina světla je absorbována prvním polarizačním filtrem, kromě toho je ale celá sestava průhledná.[9]

Podobně jako u LED mohou být LCD displeje uspořádány do segmentů nebo matic bodů. LCD technologie nesmí být trvale vystavena napájení stejnosměrným proudem. Jednotlivé body (segmenty) musí být napájeny impulsy se stejnou nebo opačnou fází. Z tohoto důvodu se LCD displeje vyrábějí jako funkční celek tvořený vlastní LCD maticí, inteligentním řadičem a systémem podsvícení. Displeje s integrovaným řadičem jsou ve dvou provedeních:

3.2.2.1 Znakové LCD

Znaková – matice LCD bodů je uspořádána do 1 až 4 řádků s 8 až 20 znaky. Jednotlivé znaky tvoří zpravidla matice 5x8 bodů. Řadič umožňuje přímé psaní znaků nebo řetězců. Obvykle je možné definovat množinu 8 až 10 uživatelských znaků. Pomocí této funkce je možné zobrazit diakritiku nebo pseudografiku (např. sloupcový graf).[9]

3.2.2.2 Grafické LCD

Grafická – matice LCD bodů je uspořádána do pole s rozměrem od 32x64 až po 256x1024. Ovládání těchto displejů bývá odlišné podle použitého řadiče a hlavně podle konstrukce vlastní LCD matice. Zobrazování grafických prvků se řeší vhodnými grafickými funkcemi. Zobrazování textu se provádí pomocí tabulky grafické interpretace znaků (*fontu*).[9]

3.2.3 Fluorescentní displeje

Fluorescentní displej je vakuový elektronický prvek, ve kterém vzniká světlo žhavení. Přivedením napětí mezi katodu a anodu tohoto zobrazovacího prvku se emituje zpravidla zelené světlo. Uspořádání jednotlivých fluorescentních bodů je podobně jako u LCD – znakové a grafické. Fluorescentní displeje se většinou vyrábějí jako funkční celek s integrovaným řadičem. Velkou nevýhodou je velká spotřeba a malá životnost. [9]

3.2.4 Elektromechanické zobrazovací prvky

Elektromechanické (*flip-flop*) zobrazovací prvky se objevují zřídka hlavně díky vysoké ceně. Základem tohoto prvku je zobrazovací terč z plastu. Jedna strana tohoto terče je barevná (zpravidla žlutozelená nebo bílá) a druhá černá. V některých aplikacích může být barevná kombinace zelená/červená. V terči je umístěn permanentní magnet. Zobrazovací terč je umístěn v magnetickém obvodu cívky. Pokud se do cívky přivádí proud jedním směrem, natočí se terč například tak, že je vidět žlutozelená barva. Pokud se přivede proud s opačnou polaritou, natočí se terč do černé strany. Po odpojení proudu zůstává terč natočen do posledního stavu. To je hlavní výhoda tohoto zobrazení. Prvek se napájí pouze při změně stavu. Druhou výhodou je viditelnost při slunečním svitu. Tyto prvky se vyrábějí ve tvaru segmentů (používají se např. v totemech u čerpacích stanic) nebo bodů (body se opět sestavují do matic – typické použití v informačních tabulích dopravních prostředků). [9]

3.2.5 TFT obrazovky

V podstatě jde o barevný LCD displej. V barevných LCD je každý pixel rozdělený do tří sub pixelů, a to červeného, zeleného a modrého (tedy *RGB*). Svítivost každého pixelu je možné řídit nezávisle na ostatních, díky tranzistorům. Jejich kombinací lze pak dosáhnout milionů barev. Barevné složky (sub pixely) je možné sestavit v různých geometriích, v závislosti na použití monitoru. V případě, že software zná geometrii monitoru, je možné zvýšit viditelné rozlišení pomocí metody sub pixel rendering. Barevné LCD rozdělujeme na pasivní STN (*Supertwist Nematic*) a aktivní TFT (*Thin-Film Transistors*). [9]

3.2.6 Možnosti zapojení segmentových a maticových LED displejů

Ze všech možných zobrazovacích jednotek, které byly popsány v předešlé kapitole, byly zvoleny jako nejvhodnější pro TimePanel sedmi segmentové a maticové LED displeje. U LED displejů je možné zobrazovat znaky na displejích v dynamickém a statickém režimů. Pro segmentové displeje byl vybrán statický režim a pro maticové displeje dynamický režim zobrazování znaků.

V původním studentském návrhu elektrického schématu, z kterého se částečně vychází v současném řešení, byly také zvoleny LED displeje, ale byly dvojbarevné a především měly velmi slabou svítivost a nevhodně zvolené režimy spínání. V současném řešení jsou zvoleny jednobarevné LED displeje s vysokou svítivostí (Obr. 7 a 8) a vhodným režimem spínání.

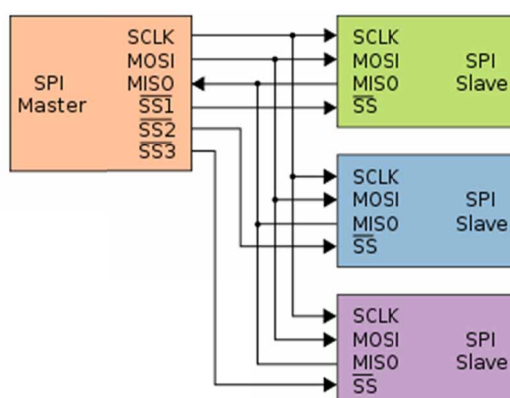
3.3 Komunikace mezi moduly v rámci zařízení (I2C, SPI)

V této podkapitole jsou popsány dostupné sběrnice (*I2C*, *SPI*) pro komunikaci mezi moduly v rámci zařízení. U jednotlivých sběrnic jsou shrnuty jejich vlastnosti, výhody a nevýhody.

3.3.1 Sběrnice SPI

Sběrnice *SPI* (*Serial Peripheral Interface*) představuje jednu z forem sériových externích sběrnic sloužících pro vzájemné propojení dvou či více komunikujících uzlů, přičemž jeden uzel obvykle vystupuje v roli takzvaného řadiče sběrnice (*master*), ostatní uzly pracují v režimu *slave*. Uzel, který pracuje jako *master*, obsahuje generátor hodinového signálu, který je rozveden do všech ostatních uzlů, čímž je umožněn zcela synchronní (navíc ještě obousměrný) přenos dat (Obr. 9).

Hodinový signál je rozváděn vodičem označovaným symbolem *SCK*. Kromě vodiče s hodinovým signálem jsou uzly propojeny dvojicí vodičů označovaných většinou symboly *MISO* (*Master In, Slave Out*) a *MOSI* (*Master Out, Slave In*), pomocí nichž se obousměrně (*full duplex*) přenáší data. Posledním signálem, který se u této sběrnice používá, je signál *SS* (*Slave Select*), jenž slouží – jak již jeho název napovídá – k výběru některého uzlu pracujícího v režimu *slave*. Maximální přenosová rychlost sběrnice je až 70MHz. Používá se pro komunikaci mezi řídicími mikroprocesory a ostatními integrovanými obvody (EEPROM, A/D převodníky, displeje...).[20]



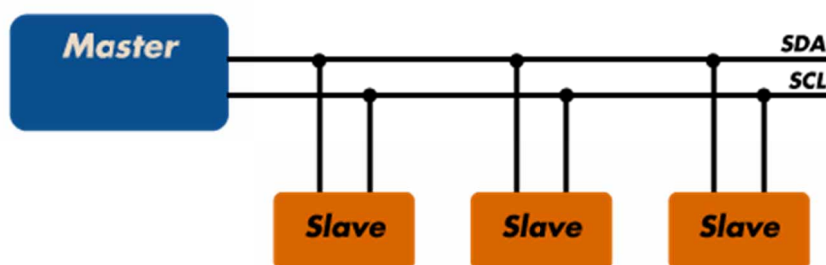
Obr. 9 Sběrnice SPI [20]

Mezi největší přednost sběrnice *SPI* patří její značná jednoduchost a tím i snadnost implementace. Jednoduchost přenosového protokolu vyplývá i z toho, že data jsou přenášena obousměrně (*full duplex*) po samostatných vodičích, takže není nutné řešit přepínání mezi vysíláním a příjmem, jako tomu bylo u některých sběrnic používajících polo duplexní přenos dat, i u dále popsané sběrnice *I2C*. [20]

Mezi nevýhody sběrnice *SPI* patří především existence pouze jednoho zařízení, které může pracovat v režimu *master*. Další nevýhodou je také nutnost použít minimálně čtyři vodiče pro obousměrnou komunikaci – *SCK*, *MISO*, *MOSI* a *GND*. A především pomocí *SPI* je možné přenášet data pouze na kratší vzdálenosti, což je dáno jednak nutností synchronizace hodinového signálu s přenášenými daty (shodné zpoždění), tak i neexistencí signálu typu *ACK* (*acknowledge*). A to je hlavní důvod proč **není** sběrnice *SPI* použita v *TimePanelu*.

3.3.2 Mezi desková komunikace - Sběrnice I2C

Je to multimasterová synchronní sériová sběrnice, která slouží pro připojení periférií k mikroprocesoru. Mohou být připojeny všechny integrované obvody, které zvládají speciální protokol sběrnice. Mimo integrovaných obvodů RAM, EEPROM, obvodů pro rozšíření portů, A/D a D/A převodníků a obvodů hodinových signálů existuje ještě celá řada speciálních integrovaných obvodů, jako například budiče displejů nebo integrovaných obvodů pro televizní a audio techniku. Obecně je možné říci, že I2C je sice poněkud složitější, ale zato flexibilnější sběrnice, která se velmi často používá i pro komunikaci na delší vzdálenosti (řádově metry), než tomu je u sběrnice *SPI*. [17] [20]



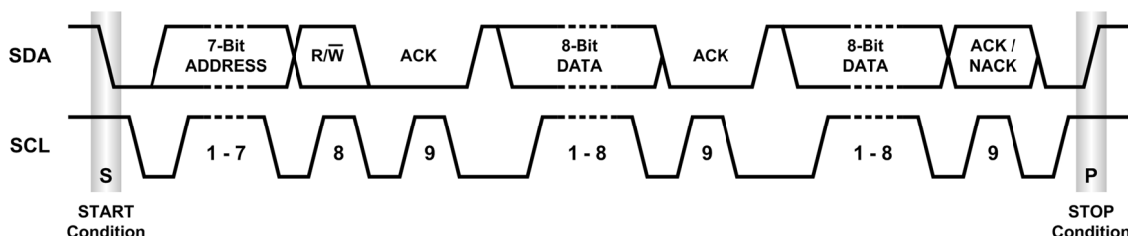
Obr. 10 Blokové schéma sběrnice s připojenými zařízeními[17]

I²C sběrnice je definována na dvou vrstvách ISO/OSI modelu:

- **Fyzická vrstva** - pro fyzické spojení se používá dvou vodičů. Maximální frekvence 100kHz nebo 400kHz. Oba vodiče (*SCL* a *SDA*) se připojují přes pull-up rezistor na napájecí napětí (Obr. 10).
- **Linková vrstva** - Master (mikroprocesor) adresuje slave (periferie) zařízení. Master může data buď zapisovat nebo číst.

Časový diagram komunikace (Obr. 11):

- Nejprve Master vyšle *START* bit, který zahajuje přenos.
- Za ním následuje 7-bitová adresa slave zařízení a jeden bit R/W (určuje, zda se bude číst nebo zapisovat z/do slave).
- Poté následuje jeden bit *ACK* (potvrzení slave, že je připraven pro přenos dat).
- Pak následuje přenos dat. Po každém přeneseném bajtu dat se vysílá bit *ACK*.
- Přenos ukončuje *STOP* bit.
-



Obr. 11 Časový diagram komunikace[20]

Protokol I2C rozeznává řadu přesně definovaných situací, které každému účastníkovi umožňují rozeznat začátek a konec přenosu a také své možné adresování:

- **Klidový stav:** *SDA* i *SCL* jsou na vysoké úrovni (HIGH) a tím neaktivní.
- **Podmínka startu:** *SDA* je masterem stažena na nízkou úroveň, zatímco *SCL* zůstává na úrovni HIGH.
- **Podmínka stopu:** i přejde z LOW na HIGH, *SCL* zůstává na úrovni HIGH. Přenos dat: Příslušný vysílač přivede na datovou linku *SDA* osm datových bitů, které jsou hodinovými impulsy na lince *SCL* vysílanými masterem posouvány dále. Přenos začíná bitem s nejvyšší vahou.
- **Potvrzení (acknowledge):** Příslušný přijímač potvrzuje příjem bytu nízkou úrovní na *SDA*, dokud master nevyšle devátý hodinový impuls na *SCL*. Potvrzení současně znamená, že se má přijímat další byte. Požadované ukončení přenosu se musí ohlásit neexistencí potvrzení. Vlastního ukončení přenosu se dosahuje podmínkou stopu.[17][20]

Sběrnice I2C0 a I2C1 jsou z řídicí jednotky TWR-PXN20 vyvedeny z primární ‚elevátorové‘ desky z pinů viz následující tabulka (Tabulka 2).

Připojené zařízení	Sběrnice	PXN20	Elevator	
Sedmi segmentové displeje	I2C0	SDA	PRIMARY	J8 - pin 8 (A8)
		SCL		J8 - pin 7 (A7)
Maticové displeje	I2C1	SDA		J9 - pin 51 (B51)
		SCL		J9 - pin 50 (B50)

Tabulka 2 Vyvedení sběrnice I2C0 a I2C1 z Freescale Tower modulu PXN20

3.4 Synchronizace času

K synchronizaci času lze použít mnoho různých způsobu. [19]

- ToE - synchronizace času přes Ethernet.
- WTD - bezdrátový systém jednotného času.
- Satelitní signál GPS.
- Rádio signál DCF 77.
- Polarizované impulsy.
- Ostatní systémy.

Z těchto způsobu řízení synchronizace času byla vybrána jako nejvhodnější, synchronizace přes Ethernet pomocí Time Serveru SNTP. Princip synchronizace je popsán níže v podkapitolách i s tabulkou použitých Time Serveru (Tabulka 3) pro aplikaci TimePanel.

3.4.1 Synchronizace s Protokoly NTP a SNTP

Existují dvě úrovně (nebo vrstvy) časových serverů protokolu NTP (*Network Time Protocol*), které jsou k dispozici na Internetu. Časové servery první úrovně jsou primárně určeny k tomu, aby fungovaly jako zdroj času pro servery druhé úrovně. Časové servery první úrovně mohou také poskytovat klíčové časové služby. K některým časovým serverům první úrovně může být omezen přístup. Časové servery druhé úrovně jsou určeny pro běžnou potřebu časových služeb protokolu SNTP a jsou obvykle veřejně přístupné. V normálních konfiguracích časových serverů protokolu SNTP se doporučuje používat časové servery druhé úrovně, protože jsou umístěné v bližší síti a mohou poskytovat rychlejší aktualizace.[18]

3.4.1.1 NTP

Je protokol z rodiny TCP/IP, který používá *User Datagram Protocol (UDP)* na portu 123. Používá se pro synchronizaci času systémů prostřednictvím paketové sítě s proměnným zpožděním a zajišťuje, aby všechny počítače v síti měly stejný a přesný čas. Byl obzvláště navržen tak, aby odolával následkům proměnlivého zpoždění v doručování paketů.

NTP klient používá Marhulův algoritmus pro stanovení času z nepatrně se lišících odpovědí časových serverů. Používá se čas UTC se speciálními příznaky pro přestupné sekundy. Běžně se jím dosahuje přesnosti hodin v řádu milisekund.[18]

3.4.1.2 SNTP

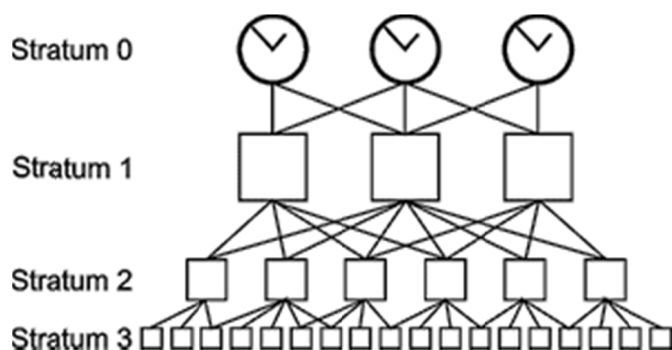
Jednodušší formou NTP a je Simple Network Time Protocol neboli SNTP. SNTP klient neuvažuje zpoždění paketů v síti a nepamatuje si stav předchozí komunikace. Používá se ve vestavěných zařízeních a v aplikacích, které nepotřebují tak extrémně vysokou přesnost.[18]

3.4.1.3 Servery poskytující synchronizaci času

Je mnoho zařízení, které nám mohou dodávat přesný čas. Nejpřesnější (referenční) jsou hodiny řízené rádiovým signálem (v ČR - DCF77) nebo přímo cesiové hodiny. Tyto zařízení označujeme jako **stratum 0**. Systémy, které od referenčních hodin bezprostředně přebírají přesný čas, jsou označovány **stratem 1**. Jednotlivé systémy tedy řadíme hierarchicky podle strata. Samotné

stratum je definované až do čísla patnáct, nicméně nejvíce časových serverů jsou se stratem 2-3. Stratum 15 má systém, který se z důvodu výpadku nemůže synchronizovat s žádným časovým serverem.

V České republice je nejvýznamnějším časovým serverem **ntp.cesnet.cz** (stratum 1), který je propojen s českým etalonem času. V rámci sítě TUONET jsou dva časové servery **ntp1.vsb.cz** a **ntp2.vsb.cz** se **stratem 2**. Tyto servery jsou dostupné i prostřednictvím celosvětové sítě Internet a svůj čas si synchronizují s časovým serverem **ntp.cesnet.cz** (stratum 1).[6][18]



Obr. 12 Hierarchické rozdělení serveru pro synchronizaci času[18]

Principiálně rozlišujeme několik základních vztahů mezi jednotlivými „účastníky“ synchronizačního procesu:

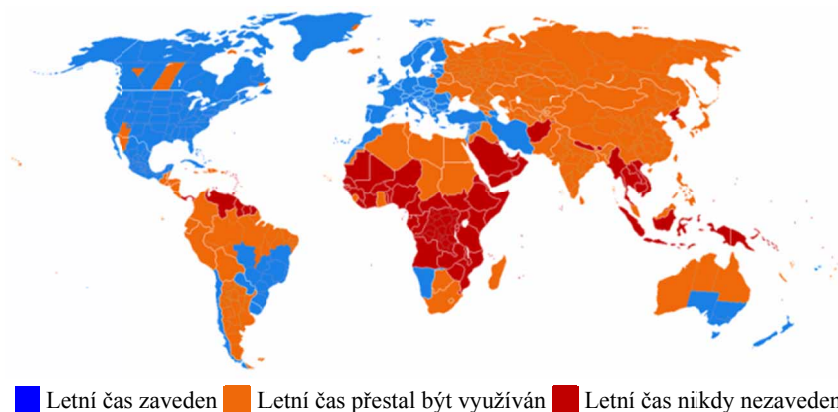
- **Server a klient** -
- **Peer**
- **Broadcast / multicast server a klient**

	Server 1	Server 2	Server 3
ServerStratum	StratumTwo	StratumTwo	StratumTwo
CountryCode	CZ	DE	UK
Hostname	time.web-systems.ws	ntp.wolfblood.net	saladfingerz.co.uk
IP Address	195.140.254.217	84.200.30.233	82.71.11.13
ServerLocation	Prague, Czech Republic	Germany	Dukinfield, Cheshire
AccessPolicy	OpenAccess	OpenAccess	OpenAccess
ServerContact	time@web-systems.ws	benjamin@benvei.at	gaz@bassbin.co.uk

Tabulka 3 Použité Time Servery v aplikaci TimePanel pro synchronizaci času

3.4.2 Letní čas – Local Time

Letní čas je označení systémové úpravy měření času, při které se v letních měsících roku nepoužívá čas daný příslušným časovým pásmem, ale používá se čas, který je o určitou hodnotu (obvykle o 1 hodinu) posunut dopředu. Cílem letního času je úspora elektrické energie, která by byla jinak potřeba pro večerní osvětlení. Důvodem je, že většina lidí je aktivnější večer (po západu slunce) než ráno (před východem slunce). Přestože je používání letního času v Evropě a Severní Americe obvyklé, většina světové populace ho nevyužívá (Obr. 13).



Obr. 13 Letní čas ve světě [21]

Na letní čas se v ČR každý rok přechází poslední neděli v březnu, kdy po 01:59:59 SEČ následuje 03:00:00. Letní čas končí poslední neděli v říjnu, kdy se po 02:59:59 SELČ hodiny posunou na 02:00:00 SEČ. Letní čas je pochopitelně záležitostí zemí mírného pásma. V tropickém pásmu se délka dne v létě od délky zimního dne neliší natolik, aby mělo zavedení letního času smysl. Obdobně v polárních oblastech, kde polární noc trvá až půl roku, nemá samozřejmě letní čas smysl.

Ve všech státech Evropy s výjimkou Islandu, Ruska, Běloruska, části Grónska a norských ostrovů Jan Mayen a Špicberky se letní čas používá. Podle pravidel EU přechod nastává ve všech časových pásmech současně, v 1:00 UTC (tzn. 2:00 SEČ, resp. 3:00 SELČ), poslední neděli v březnu, resp. říjnu.[21]

3.4.2.1 Časová pásma

Od roku 1884 bylo zavedeno 24 časových pásem (vždy po 15 stupních). Pro zjednodušení nekopírují zóny přímo poledníky, ale většinou hranice států. Některé státy však zahrnují více časových zón. Situaci ještě komplikuje nejednotné používání letního času (*Daylight Saving Time - DST*). Další důležitou zkratkou je *UTC*. Ten je odvozen od GMT, tedy lokálního času na hvězdárně v Greenwichi (nedaleko Londýna), kudy prochází nultý poledník.[21]

3.4.2.2 Jak získat „Local time“?

iCalendar je standard (*RFC 5545*) pro výměnu kalendářových dat. Je též znám pod zkratkou *iCal*. Standard vznikl kvůli potřebě poskytovat otevřenou výměnu kalendářních a plánovacích informací (úkolů a schůzek mezi uživateli) v rámci internetu. Soubor *iCalendar* má příponu *ics* a data v něm jsou tvořena prostým textem (*ASCII*). Délka každého řádku v souboru může být maximálně 75 bytů (ne znaků) a je ukončen znakem *CRLF* (*hex: 0D0A*).

Všechna data v souboru jsou umístěna v objektu kalendáře (*Calendaring Core Object*). První a poslední řádek objektu má striktně danou podobu. Nevýhodou icalendaru, je, že má poměrně striktní formátování a výsledný soubor je pro člověka vcelku nepřehledný. Konkrétní prvky kalendáře jsou vyjádřeny pomocí komponent, kdy každá komponenta začíná písmenem „V“.[22]

Komponenty kalendáře:

- To-do - úkol
- Event - událost - tu dnes popíšu
- Journal - deník
- Free/Busy - požadavek na volný čas
- TimeZone - časová zóna
- Alarm – upomínky

S těchto komponent byla vybrána jako vhodná komponenta pro aplikaci TimePanel *VTIMEZONE*, která je popsána níže. V aplikaci TimePanel se tak letní čas („Local time“) nastaví buď automaticky pomocí komponenty *VTIMEZONE* nebo manuálně pomocí příkazového řádku *SHELL* a příkazu „time_offset_set“.

3.4.2.2.1 Časová zóna (VTIMEZONE)

VTIMEZONE komponenta popisuje časovou zónu kalendáře. Definiuje standardní a zimní čas pro dané časové pásmo a časový interval. *VTIMEZONE* nesmí být vnořený do žádné jiné komponenty, ale musí obsahovat *TZID* a také alespoň jednu definici času *STANDARD* nebo *DAYLIGHT*. Tyto definice musí zahrnovat *DTSTART* *TZOFFSETFROM* a *TZOFFSETO* vlastnosti. V jenom kalendáři může být více *VTIMEZONE*, ale pak musí mít unikátní *TZID*. Tato možnost je nezbytná například pro letecké společnosti, kdy letadla startují v jedné zóně a přistávají v jiné.[22]

4 Realizace navrženého řešení

V této kapitole je zdokumentovaná realizace navrženého řešení TimePanelu. Tato kapitola je rozdělná do tří jednotlivých částí. První, mechanická část je věnovaná návrhu a realizaci konstrukce TimePanelu a uspořádání jednotlivých modulu v konstrukci. Druhá část je zaměřena na návrh a realizaci DPS modulu TimePanelu a také na výběr hardwarových součástí. V třetí, softwarové části jsou popsány nejpoužívanější příkazy pro příkazový řádek, režimy přepínání lokalit u TimePanelu a na závěr kompletní aplikace pro zařízení s použitím UML diagramu.

4.1 Mechanická část – konstrukce

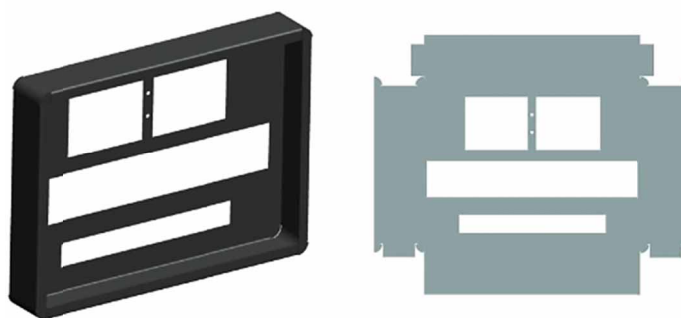
V rámci studentských projektů byly vytvořeny předběžné konstrukční návrhy konstrukce TimePanelu. Konečný návrh řešení konstrukce vychází z těchto podkladů a tyto návrhy vylepšuje a rozšiřuje. V této podkapitole jsou tedy popsány studentské konstrukční návrhy, uspořádání LED modulů v konstrukci a návrh a realizace konečného řešení konstrukce TimePanelu.

4.1.1 Konstrukční návrhy TimePanelu

Podle seznamu požadavků pro zařízení, byly navrženy čtyři konstrukční varianty řešení TimePanelu, ze kterých bude vybrána jedna varianta, podle které bude konstrukce TimePanelu navržena (detailně) a zrealizována. U zvolené varianty budou provedeny drobné úpravy a bude pro ni vytvořena technická zpráva včetně finanční rozvahy, výrobních výkresů (AutoDesk Inventor), 3D modelů a foto dokumentace.

4.1.1.1 Plechový skelet

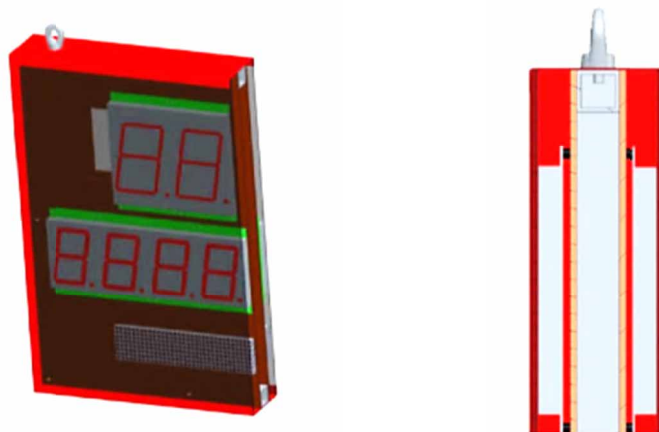
Skelet je tvořen dvěma plechovými výlisky, do kterých jsou zamontovány LED displeje. Jelikož se jedná o ocelový plech, stává se celé řešení z hlediska hmotnosti těžké a po stránce vzhledu neestetické, protože lze vidět zbytky po svařování skeletu a také spojovací materiál na přední straně (Obr. 14).



Obr. 14 Návrh 1 - plechový skelet

4.1.1.2 Dva oddělené panely

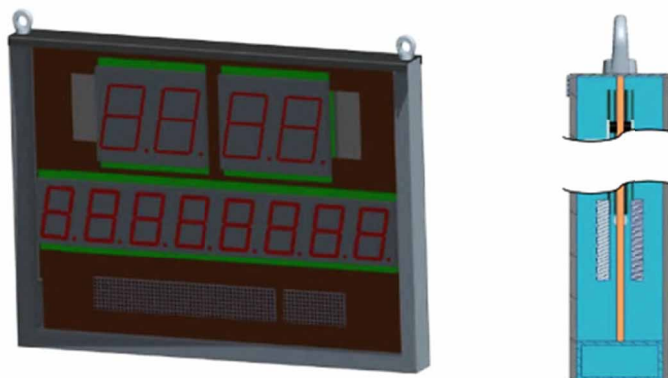
Základ tvoří rám z čtvercových profilů spojených pomocí dvou desek s LED panely. Ochranné plexiskla ukotvují čtyři plechové kryty ve tvaru U, které jsou přišroubovány k rámu z profilů. Spojovací materiál je nyní vidět pouze z boku, avšak je nutné použít dvou samostatných desek pro upevnění LED panelů (Obr. 15).



Obr. 15 Návrh 2 - dva oddělené panely

4.1.1.3 Oboustranný panel zasunutý do svařence

Oproti předchozí variantě je použita jedna nosná deska s oboustranným osazením LED displejů. Tato deska je ze shora zasunuta do svařence tvaru U tvořeného z obdélníkových profilů. Plexiskla jsou na svařenec nalepena a z horní strany překlenuta plechovým krytem přišroubovaným k rámu. Konstruktivní materiál je vidět pouze shora. Toto řešení vyžaduje výrobní technologii svařování. Dojde-li k poškrábání plexiskla, není možná jeho výměna (Obr. 16).



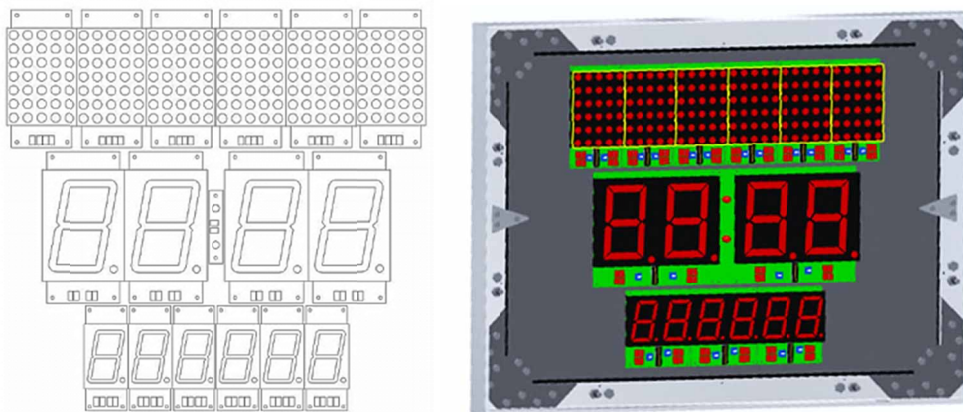
Obr. 16 Návrh 3 - Oboustranný panel zasunutý do svařence

4.1.1.4 Šroubovaná konstrukce s oboustranným panelem

Nosným prvkem této varianty je rám tvořený čtyřmi T profily, které se vzájemně sešroubují. Na tento rám je upevněna nosná deska oboustranně osazená LED displeji. Díky sešroubování všech dílů je konstrukčně nejjednodušší a použitím T profilů není vidět spojovací materiál. Toto řešení bylo svými vlastnostmi vybráno jako vítězné a v dalších kapitolách bude detailně popsáno.

4.1.2 Uspořádání LED modulů v konstrukci

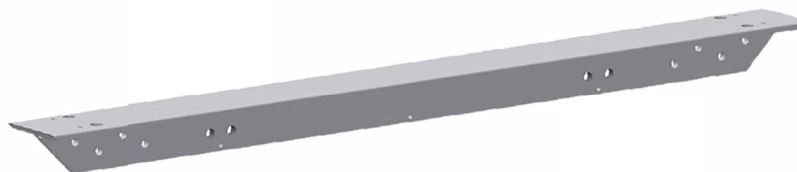
První řádek je tvořen maticovými displeji pro zobrazení názvu města, druhý pro čas a poslední pro datum. Název města je zobrazen 6 písmeny, pro čas jsou standardně vyhrazeny 4 displeje (2- hodiny, 2- minuty) a pro datum je použito 6 displejů (2 – dny, 2 – měsíce, 2 – roky). Jejich rozložení je na následujícím obrázku (Obr. 17)



Obr. 17 Uspořádání LED modulu (lokality, čas, datum) a 3D model otevřeného TimePanelu

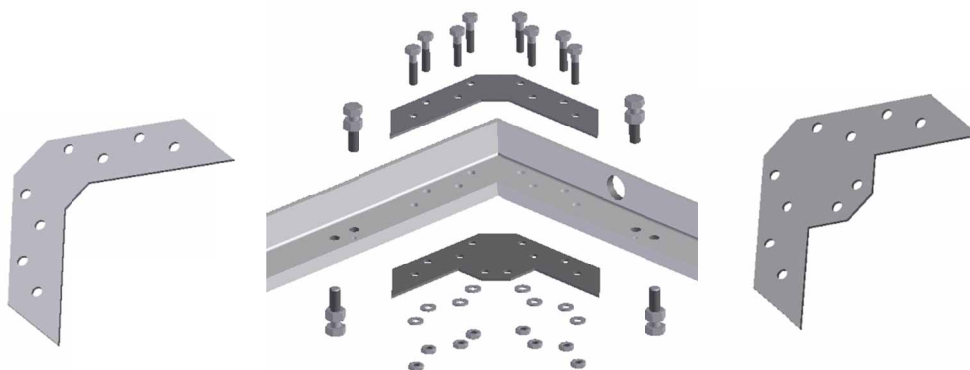
4.1.3 Konečné řešení konstrukce TimePanelu

Ze všech čtyř navrhovaných konstrukčních řešení byla vybrána jako nejvhodnější varianta šroubovaná konstrukce s oboustranným panelem. Jak již bylo řečeno, rám tvoří čtyři profily tvaru T s rozměry 60x40x6 mm které se vzájemně sešroubují. Jelikož rám je obdélníkového tvaru, budou zapotřebí 2 ks o délce 720 mm (Obr. 18) a 2 ks o délce 540 mm. V konstrukci jsou použity profily z hliníku, protože váží čtyři krát méně než ocelové profily. Celková hmotnost sestavené konstrukce, bez středové desky a plexiskel, činí zhruba 6 kg. Tento AL profil je dodáván firmou ProAL s.r.o. s pořizovací cenou 1000 Kč. Vytvoření návrhu konstrukce a výkresové dokumentace pro její realizaci trvalo tři měsíce. Výkresová dokumentace a 3D model TimePanelu byl vytvořen v programu AutoDesk Inventor 2013. Obrobení profilů podle výkresové dokumentace konstrukce (Příloha VII) je provedeno firmou **ContiTherm s.r.o.** s odhadovanou cenou 4000Kč.



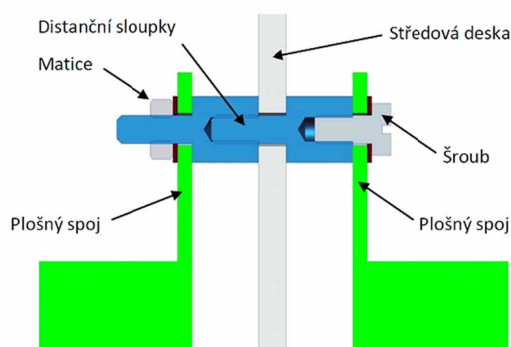
Obr. 18 Horní hliníkový profil délky 720mm

Hliníkové profily jsou spojeny pomocí konzol z hliníkového plechu tloušťky 2mm a spojovacího materiálu (Obr. 19). Konzole z hliníkového plechu byly vyrobeny opět dle výkresové dokumentace (Příloha VII) firmou **ContiTherm s.r.o.** s odhadovanou cenou 3000Kč. Jedná se o dva druhy konzol, každá po 4 ks viz obrázky (Obr. 19). Cena spojovacího materiálu potřebného pro sestavení konstrukce jako šrouby, matice a podložky je zhruba 800 Kč viz příloha (Příloha VI). Na všech částech konstrukce byla provedena povrchová úprava pískováním v stříbrné barvě, firmou **ContiTherm s.r.o.**



Obr. 19 Spojení hliníkových profilů pomocí konzolí z hliníkového plechu a spojovacího materiálu

Středová deska, která nese všechny moduly TimePanelu, je tvořena z tvrdého plastu (*Medur*) o rozměrech 635x455 mm a je uchycena ke konzolám pomocí spojovacího materiálu. Středová deska z tvrdého plastu dodávána firmou Řempe je vlastnoručně obrobena a její cena se pohybuje okolo 1400 Kč.



Obr. 20 Montáž modulu TimePanelu na středovou desku

Jednotlivé moduly TimePanelu jsou namontovány na středovou desku pomocí kovových distančních sloupků dodávaných firmou GME s.r.o. (Obr. 20). Každý modul s displejem je uchycen ve čtyřech bodech. Jelikož LED displeje u modulu mají různou výšku, jsou použity sloupky s délkou 5 a 8 mm. Pod šroub a matici je potřeba vložit izolační – plastovou podložku. Celková pořizovací cena spojovacího materiálu pro moduly je 800 Kč.



Obr. 21 Způsob zavěšení TimePanelu

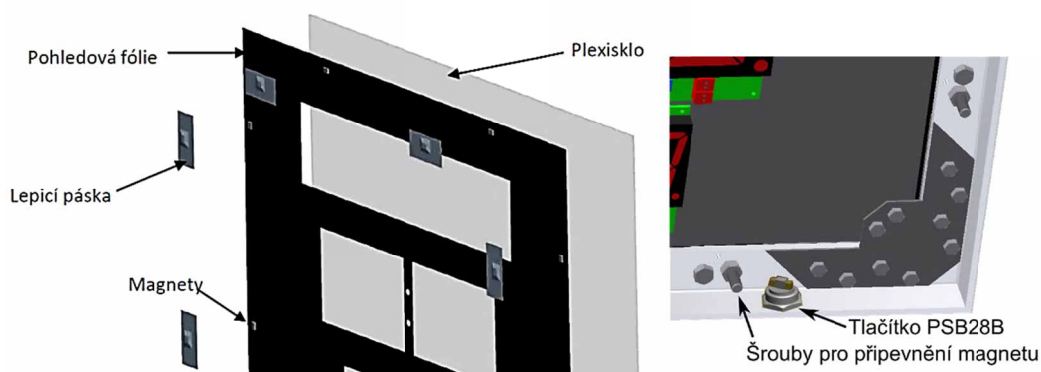
Celý TimePanel je zavěšen pomocí nerezového lanka o průměru 2 mm a lankové svorky DIN741. Nerezové lanko je protaženo malými dírkami v horním profilu rámu a ukotveno pomocí lankové svorky DIN741 viz obrázek (Obr. 21). Moduly na obou stranách TimePanelu jsou zakryty pomocí

čirého plexiskla s polepovou folii, z důvodů zvýšení odolnosti panelu proti vniknutí cizího tělesa či vniknutí kapaliny, ale také z vizuálního hlediska. Čiré plexisklo Akrylon XT s rozměry 706x526 mm a tloušťkou 3 mm bylo vlastnoručně obrobena a je dodáváno firmou Řempe s.r.o. za pořizovací cenu 930 Kč.



Obr. 22 Neodýmové magnety 10x7x2 mm

Na plexisklo je nalepena z vnitřní strany černá pohledová fólie, která zakryje ostatní elektroniku, a přes otvory v ní budou viditelné pouze displeje. Samolepící fólie je nařezaná podle umístění (pozice) displejů a rozměrů plexiskla. Její pořizovací cena je 300Kč. Na tuto fólii jsou ze zadní strany upevněny lepicí páskou neodýmové magnety dodávané firmou Selos, pomocí kterých drží plexisklo v konstrukci. Plexisklo je pomocí magnetů připevněno na hlavičky šroubů M6 zašroubovaných v rámu TimePanelu viz obrázek (Obr. 23). Rozměry magnetu jsou 10x7x2 mm se silou 9,8 N (Obr. 22). Cena potřebných 32 magnetů je 880 Kč.



Obr. 23 Plexisklo s pohledovou folii a magnety a umístění tlačítka v konstrukci

Na plexiskle je navíc na vnější straně nalepena černá proti sluneční polepová fólie, **která** slouží jako ochrana proti poškrábání plexiskla a především k tomu aby byly vidět na displejích pouze aktivní – svítící segmenty (tečky). Na předešlém obrázku (Obr. 23) je vidět umístění tlačítka pro přepínání lokalit TimePanelu. Jsou použity dvě tlačítka, každé pro jednu stranu displejů, dodávané firmou GME s.r.o. s označením PSB28B. Jejich cena je 132 Kč. Všechny materiály a součásti použité v konstrukci jsou sepsány i s cenami v příloze (Příloha VI). Celková velikost TimePanelu je **720 x 540 x 60 mm** s celkovou hmotností (bez elektroniky) 12.5KG (Obr. 24).



Obr. 24 Konstrukce TimePanelu bez elektroniky

Další fotografie TimePanelu a jeho částí konstrukce jsou v příloze (Příloha I).

4.2 Návrh a realizace schémat a DPS

Tato kapitola popisuje návrh schémat modulů a desek TimePanelu pomocí *Formica Schematic* se všemi potřebnými výpočty předřadných odporů, ztrátových výkonů atd. V kapitole budou částečně popsány i společné prvky (součástky) modulu TimePanelu a také jak bude realizováno propojení mezi jednotlivými moduly. Dále budou z vytvořených schémat vygenerovány netlisty a následně z nich navrženy plošné spoje podle základních pravidel a zásad návrhu plošných spojů ve *Formica Layout*. Při návrhu byly kombinovány všechny tři metody návrhu DPS (metoda schématu, konektorů a centrální součástky). V programu *Formica Layout* a *Schematic* bylo potřeba vytvořit knihovnu s pouzdry a schematickými značkami, které původní knihovna neobsahovala.[1][2][3]

V původním studentském návrhu schémat a DPS, z kterého se částečně vychází v současném řešení, nebyly vůbec spočteny hodnoty předřadných odporů displejů, ztrátové výkony ani spotřeba modulů. Dále v návrhu u DPS byly použity pouze součástky pro klasickou montáž THT a proto byly DPS příliš velké a nevlezly se do zadaných rozměrů konstrukce. Z těchto důvodů byly navrženy nové schémata a DPS modulů s vhodnými součástkami a rozměry pro TimePanel.

4.2.1 LED panely – moduly Time panel 1 až 3

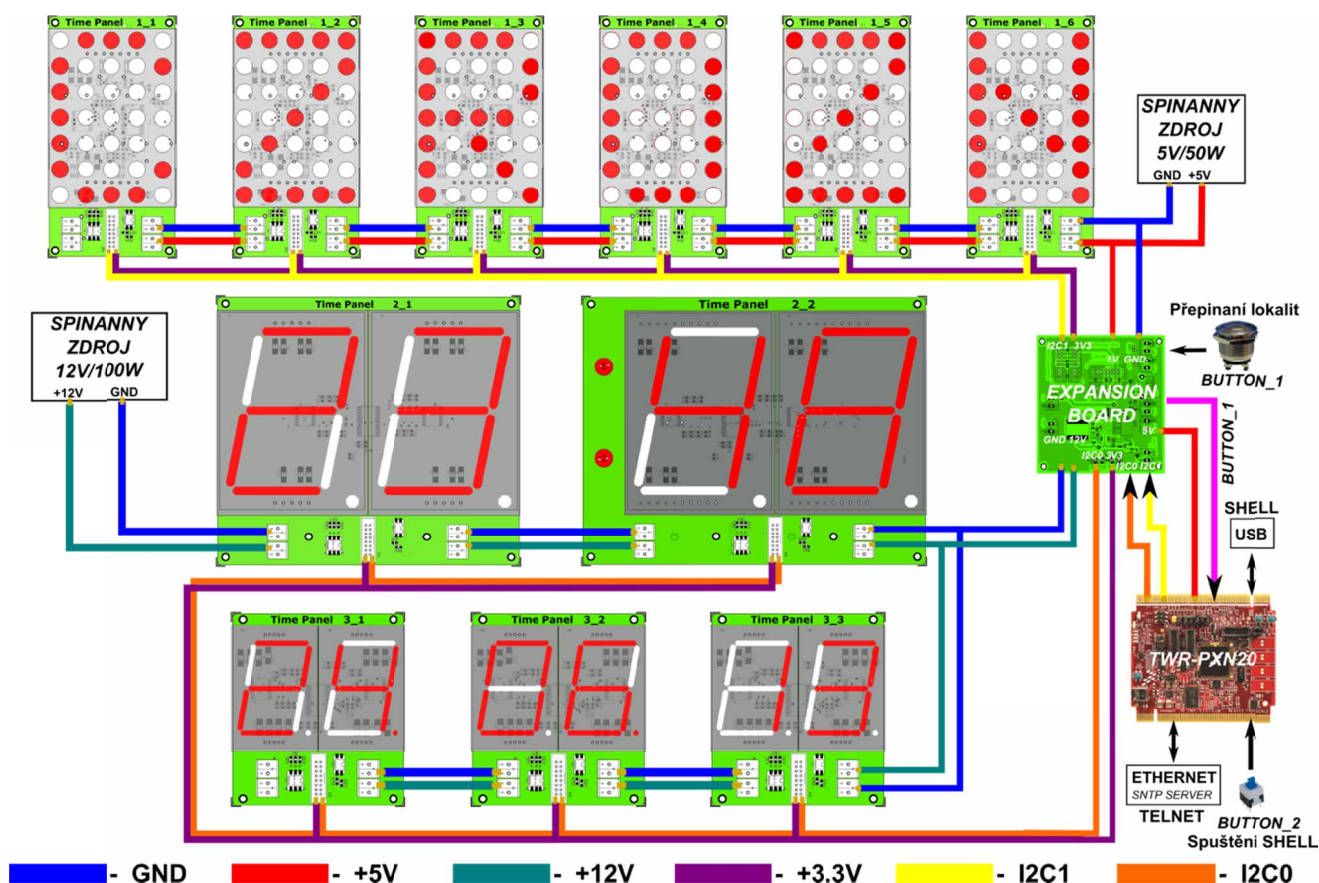
Jak už bylo řečeno (viz. Kapitola 2), TimePanel slouží pro zobrazení světového času a data ve vybraných lokalitách s různými časovými pásmy. Pro zobrazení času, data a lokality jsou použity jednobarevné sedmi segmentové a maticové LED displeje s vysokou svítivostí a rozdílnými velikostmi a zapojením. A proto bude muset TimePanel obsahovat tři odlišné moduly.

Modulů TimePanelů bude zapotřebí několik:

- **Velký:** zobrazení hodin – **TimePanel 2**
- **Menší:** zobrazení data – **TimePanel 3**
- **Maticový:** zobrazení názvu města – **TimePanel 1**

Ze všech možných nabízených zobrazovacích jednotek byly vybrány jako nejvhodnější pro TimePanel, zobrazovací jednotky na principu LED diod. Pro zobrazení času a data byly zvoleny sedmi segmentové LED displeje a pro zobrazení lokality byly vybrány maticové LED displeje (5 x 7).

Realizace propojení mezi jednotlivými moduly a také řídicí jednotkou je detailně popsána v kapitole (Kapitola 5.2.2) a také v blokovém schématu TimePanelu viz obrázek (Obr. 25). Na všech modulech TimePanelu bude použita stejná skupina součástek (společné součástky), nehledě na to, že moduly mají různé rozměry a jsou použity jiné typy displejů. Společné součástky jsou popsány v podkapitole (viz. Kapitola 5.2.3).



Obr. 25 Blokové schéma TimePanelu

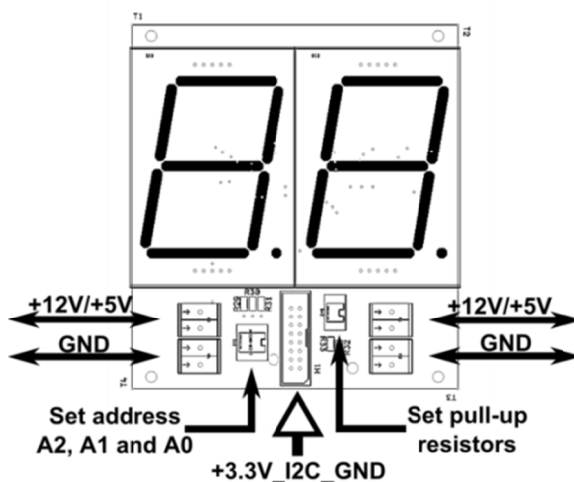
4.2.2 Propojení modulů TimePanelu

Propojení jednotlivých desek TimePanelu je rozděleno do dvou částí. V první části bude realizováno propojení desek pomocí 16-žilového plochého kabelu 0,14mm s konektory *MLW16G* a *PFL16*. Tímto plochým kabelem jsou napájeny jednotlivé expandéry napětím 3.3V a také jsou kabelem vedeny datové signály pro sběrnici I2C. Jelikož jsou pomocí plochého kabelu napájeny expandéry, je z důvodu nízké proudové zatížitelnosti plochého kabelu nutno použít více vodičů plochého kabelu pro 3.3V (6 vodičů) a GND (6 vodičů).



Obr. 26 Konektory (PFL16 a ARK300) a DIP přepínač pro nastavení adresy expandérů

Druhá část propojení je realizována pomocí měděných vodičů a konektorů *ARK300-2* a slouží k napájení jednotlivých displejů TimePanelu pomocí 5V (TimePanel 1) a 12V (TimePanel 2 a 3). Při volbě vodičů a návrhů vodičových cest na DPS je zapotřebí počítat s jejich proudovou zatížitelností a podle ní je zapotřebí zvolit průměr vodičů a šířku vodičových cest DPS v TimePanelu. Pro výpočet proudové zatížitelnosti plošných spojů byl použit software.



Obr. 27 Struktura rozmístění konektoru na modulech TimePanelu

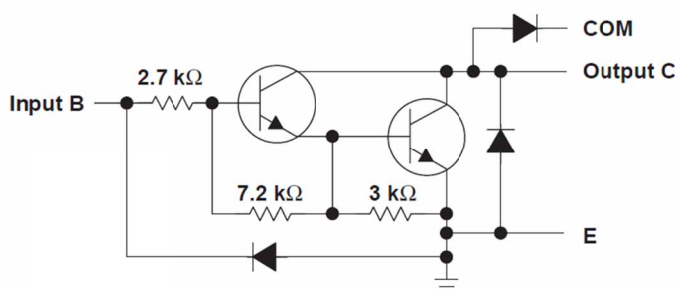
Všechny moduly obsahují DIP přepínače pro hardwarové nastavení adresy expandérů a nastavení pull-up odporu na I2C. U všech modulů TimePanelu (TimePanel 1 až 3) je struktura rozmístění konektorů a přepínačů DIP stejná a to z důvodů přehledného a snadného propojování jednotlivých modulů a snadné manipulaci s moduly (nastavení adresy) viz obrázek (Obr. 27).

4.2.3 Společné součástky pro moduly TimePanelu

V této kapitole jsou popsány vlastnosti společných součástek obsažených na každém modulu TimePanelu 1 až 3. V původním návrhu elektrického schématu byl zvolený expandér pro napájecí napětí 5V a bylo nutné převádět napětí sběrnice I2C z 3.3V na 5V. Proto je v současném řešení vybrán expandér MCP23016 s provozním napájecím napětím 3.3V.

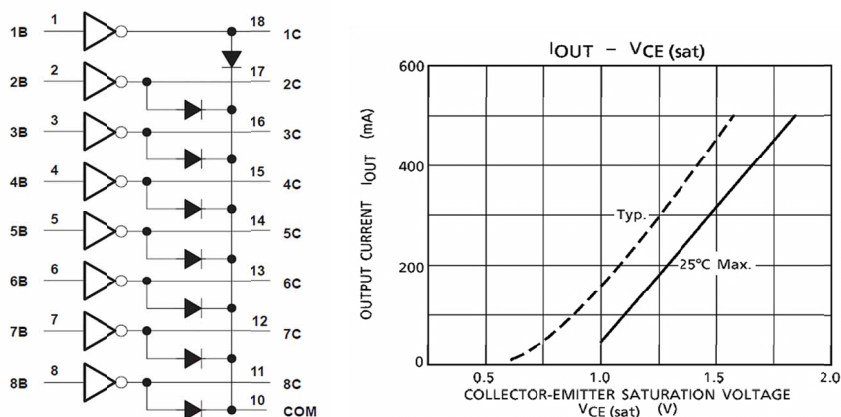
4.2.3.1 Tranzistorové pole ULN2803A

Tranzistorové pole ULN2803A se skládá z osmi NPN Darlingtonových dvojic, které představují napěťové výstupy se společnou katodou diody pro spínání indukčních zátěží. Maximální kolektorový proud tekoucí každým párem je 500 mA (Obr. 28). Darlingtonové páry mohou být připojeny paralelně pro vyšší proudové schopnosti obvodu.[5][6]



Obr. 28 Schéma Darlingtonová zapojení každého kanálu UNL2803A [5][6]

Obvod ULN2803A má 2.7kΩ bázevý odpor pro každý vstup pro provoz přímo s TTL nebo CMOS zařízením (Obr. 29). Tyto obvody jsou určeny ke spínání větších proudů a napětí než zvládnou běžné digitální obvody či mikrořadiče. [5][6]



Obr. 29 Logické uspořádání tranzistorového pole a závislost úbytku napětí mezi C a E- $V_{CE(sat)}$ na I_{out}

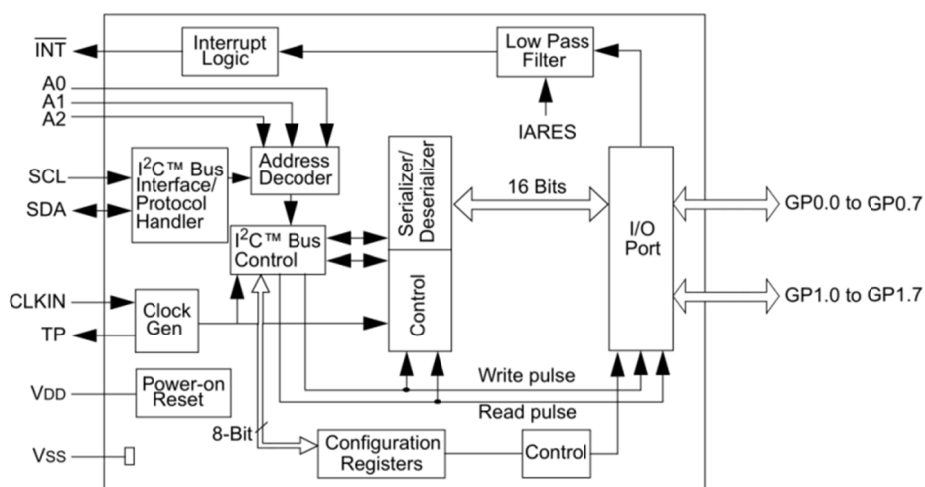
4.2.3.1.1 Vlastnosti ULN2803A:

- 500 mA maximální proud tekoucí kolektorem (pro jeden kanál)
- maximální spínané napětí 50 V
- výstupní clamp dioda pro spínání indukčních zátěží
- vstupy kompatibilní s různými typy logiky (TTL, CMOS)
- maximální ztrátový výkon 1.47W
- pouzdra DIP18 a SOL18

4.2.3.2 Šestnáctibitový I/O expandér I²C - MCP 23016

Tento obvod sdružuje 2 x 8 bitový obousměrný GPIO port a linky jsou defaultně nastaveny jako vstup. GPIO linky mají vysokou proudovou zatížitelnost (+/- 25 mA; typ OC) - čili je možno s nimi přímo budit např. LED a podobně. MCP23016 je určen pro rozsah napájecího napětí 2,0 až 5,5 V a rozsah pracovních teplot -40 až 85 °C. Obvod podporuje standardní rozsah kmitočtů 0 - 400 kbps. Jako zdroj interního kmitočtu 1 MHz slouží jednoduchý RC člen (3k9, 33pF), který se jednoduše zapojí na vstup CLK. MCP23016 obsahuje výstupní linku přerušení a tři vstupní hardwarové adresovací linky (A0, A1, A2).[12]

Jeden I2C expandér je schopen obsloužit pouze 2 sedmi segmentové nebo 1 maticový LED displej. Pro jednu stranu TimePanelu je potřeba 11 kusu expandérů, ale jelikož TimePanel je oboustranný, je použito celkem 22 kusu.



Obr. 30 Blokové zapojení obvodu MCP23016 [12]

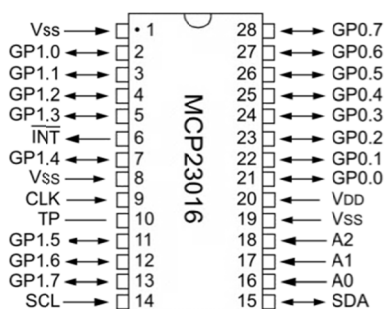
4.2.3.2.1 Vlastnosti MCP23016 :

- 16-bit vzdálený obousměrný I/O port (výchozí stav 16 vstupů)
- Rychlá I2C[™] sběrnice s taktovací frekvencí (0 až 400 kbit/s)
- Tři adresovací piny umožňují použití až osm zařízení
- Výstupní proud na kanál až ±25 mA
- Vnitřní Power-On Reset (POR)
- Registr pro konfiguraci inverze polarity údajů vstupních portů
- Kompatibilní s většinou mikrokontrolérů
- K dispozici teplotní rozsah -40 °C až +85 °C
- Provozní napájecí napětí 2,0 až 5.5V
- Nízký pohotovostní proud

U obvodu se nastavuje režim V/V linek pomocí registru **IODIR0** nebo **IODIR1**(IO direction) pro port 0 nebo port1, zapsáním úrovně log. 1 do příslušného bitu nastavíme bit do vstupního režimu, zapsáním log. 0 jej nastavíme do výstupního režimu.

Vlastní výstupní registry jsou **GP0, GP1** (general purpose IO port) jejichž hodnoty jsou po zápisu do těchto registrů kopírovány do brány (opět **GP0 = PORT0** - čili **GP0.0 až GP0.7** a obdobně pro GP1). Při zápisu do těchto V/V registrů se hodnota zapisuje také do výstupních latch registrů (OLATH).

Obvod má také výstupní registry (**OLATH0 a OLATH1 - output latch register**). Tyto registry obsahují aktuální hodnotu, zapsanou do výstupu. Pokud tedy odešleme hodnotu na port (zápis do GP0,GP1) hodnota je zapsána do registru a až pak je nastavena hodnota na výstupní bráně. Pokud čteme hodnotu tohoto registru (OLATH), pak čteme hodnotu naposledy zaslanou na bránu, ne hodnotu, která je skutečně při čtení na bráně. Skutečnou hodnotu na bráně získáme, čteme-li registr GP0 (GP1).



Command Byte	Result
0h	Access to GP0
1h	Access to GP1
2h	Access to OLAT0
3h	Access to OLAT1
4h	Access to IPOL0
5h	Access to IPOL1
6h	Access to IODIR0
7h	Access to IODIR1
8h	Access to INTCAP0 (Read-Only)
9h	Access to INTCAP1 (Read-Only)
Ah	Access to IOCON0
Bh	Access to IOCON1

Obr. 31 Pouzdro obvodu MCP23016 a Command byte pro MCP23016 [12]

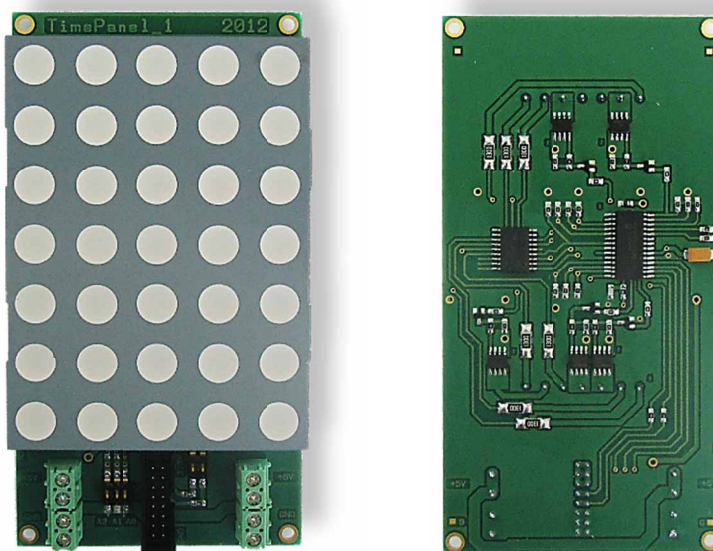
Registry **IPOL0, IPOL1** (input polarity) jsou registry pro nastavení polarity **vstupních signálů**. Pokud je bit v tomto registru nastaven, pak je hodnota odpovídajícího bitu v GP0 (GP1) invertována (tedy hodnota, která je v okamžiku čtení na odpovídajícím bitu), pokud nastaven není, čteme skutečnou hodnotu na odpovídajícím vývodu. Registry **INTCAP0, INTCAP1** (interrupt capture) jsou registry, které obsahují po přerušení poslední načtenou hodnotu na portu.

Registry **IOCON0, IOCON1** (IO expander control). Každý z těchto registrů obsahuje pouze jeden nastavitelný bit - **IARES** (Interrupt Activity Resolution). Tímto bitem lze nastavit smplovací rychlost expandéru, tzn. rychlost, s jakou se u vstupního režimu testují změny na portu pro vyvolání přerušení při změně hodnoty. Pokud je tento bit nastaven (log. 1) pak je rychlost smplování 200 μ s a obvod odebírá vyšší proud. Pokud je (implicitně) bit nulován, je rychlost smplování 32 ms a odběr obvodu je nižší.

Obvod dále umožňuje využít výstup přerušení, který je společný pro oba porty. Tento výstup je výstup s otevřeným kolektorem. Po změně na portu je výstup INT stažen do nízké úrovně.[12]

4.2.4 Modul TimePanel_1

Modul TimePanel 1 slouží k zobrazení názvu (1 znaku) jednotlivých světových lokalit. Deska obsahuje jeden I2C expandér MCP23016 pro jeden **5x7 maticový displej TA40-11SRWA**. Celkově TimePanel obsahuje dvanáct desek TimePanel_1. Rozměry jedné desky TimePanelu_1 jsou **74.725x142.500 mm** (Obr. 32). Rozměry, zapojení a technické parametry 5x7 maticového displeje TA40-11SRWA jsou v příloze (Příloha II). Modul TimePanel_1 byl původně navržen pro napájení z 12V zdroje, ale z důvodu velkých ztrátových výkonů na odporech byl tento návrh pozměněn a modul je navržen pro napájení z 5V zdroje.



Obr. 32 Modul TimePanel_1 (TOP a BOTTOM)

4.2.4.1 Výpočty předřadného odporu a ztrátových výkonů tečky (DOT) pro napájecí napětí $U_{NAPAJECI} = 12V$

$$U_{NAPAJECI} = 12V$$

$$V_{F(typ)} = 1.85$$

$$I_{F(typ)} = 20mA$$

$$V_{CE(sat)} = 0.6V \quad - \text{viz graf } V_{CE(sat)} = f(I_{OUT})$$

$$U_{RLED} = U_{NAPAJECI} - V_{F(typ)} - V_{CE(sat)} = 12 - 1.85 - 0.6 = 9.55V$$

$$P_{LED} = V_{F(typ)} \cdot I_{F(typ)} = 1.85 \cdot 0.02 = 37mW$$

Předřadný odpor R_{LED} :

$$R_{LED} = \frac{U_{RLED}}{I_{F(typ)}} = \frac{9.55}{0.02} = 478\Omega$$

$$P_{RLED} = U_{RLED} \cdot I_{F(typ)} = 9.55 \cdot 0.02 = 191mW$$

Výkon potřebný pro jednu tečku maticového displeje s $R_{LED} = 478\Omega$:

$$P_{DOT} = P_{RLED} + P_{LED} + P_{TRAN} = 191 + 37 + 10 = 238mW$$

Jelikož sloupec maticového displeje obsahuje sedm řádku (teček), P_{DOT} je nutno vynásobit sedmi.

$$P_{SLOUPCE} = P_{DOT} \cdot 7 = 230mW \cdot 7 = 1.61W \cong 1.7W$$

Výkon $P_{SLOUPCE}$ je maximální výkon který může displej odebírat ze zdroje, protože je spočtený pro stav, kdy budou svítit všechny tečky (řádky) v daném sloupci maticového displeje. Ve skutečnosti však najednou všechny tečky sloupce displeje svítit nemusí a proto bude odebíraný výkon nižší.

Modul TimePanel_1 obsahuje maticové displeje TA40-11SRWA a jeden expandér MCP23016 který má maximální ztrátový výkon $P_{EXPANDER} = 1W$ (300mA, 3.3V). Proto maximální odebíraný výkon TimePanelu_1 je:

$$P_{TimePanel_1} = P_{SLOUPCE} + P_{EXPANDER} = 1.7 + 1 = 2.7W$$

Celý TimePanel obsahuje dohromady dvanáct TimePanelu_1 (1x TA40-11SRWA a 1x MCP23016). Celkový maximální odebíraný výkon všech TimePanelu_1 pro zobrazení názvu lokality v celém zařízení je zhruba:

$$P_{TimePanel_1_ALL} = 12 \cdot P_{TimePanel_1} = 2.7 \cdot 12 = 32.4 \cong 33W$$

4.2.4.2 Výpočty předřadného odporu a ztrátových výkonu tečky (DOT) pro napájecí napěti $U_{NAPAJECI} = 5V$

$$U_{NAPAJECI} = 5V$$

$$V_{F(typ)} = 1.85$$

$$I_{F(typ)} = 20mA$$

$$V_{CE(sat)} = 0.6V \quad - \text{ viz graf } V_{CE(sat)} = f(I_{OUT})$$

$$U_{RLED} = U_{NAPAJECI} - V_{F(typ)} - V_{CE(sat)} = 5 - 1.85 - 0.6 = 2.55V$$

$$P_{LED} = V_{F(typ)} \cdot I_{F(typ)} = 1.85 \cdot 0.02 = 37mW$$

Předřadný odpor R_{LED} :

$$R_{LED} = \frac{U_{RLED}}{I_{F(typ)}} = \frac{2.55}{0.02} = 128\Omega$$

$$P_{RLED} = U_{RLED} \cdot I_{F(typ)} = 2.55 \cdot 0.02 = 51mW$$

Výkon potřebný pro jednu tečku maticového displeje s $R_{LED} = 128\Omega$:

$$P_{DOT} = P_{RLED} + P_{LED} + P_{TRAN} = 51 + 37 + 10 = 98mW \cong 100mW$$

Jelikož sloupec maticového displeje obsahuje sedm řádku (teček), P_{DOT} je nutné vynásobit sedmi.

$$P_{SLOUPCE} = P_{DOT} \cdot 7 = 100mW \cdot 7 = 700mW \cong 0.7W$$

Výkon $P_{SLOUPCE}$ je maximální výkon který může displej odebírat ze zdroje, protože je spočtený pro stav kdy budou svítit všechny tečky (řádky) v daném sloupci maticového displeje. Ve skutečnosti však najednou všechny tečky sloupce displeje svítit nemusí a proto bude odebíraný výkon nižší.

Modul TimePanel_1 obsahuje maticové displeje TA40-11SRWA a jeden expandér MCP23016, který má maximální ztrátový výkon $P_{EXPANDER} = 1W$ (300mA, 3.3V). Proto maximální odebíraný výkon TimePanelu_1 je:

$$P_{TimePanel_1} = P_{SLOUPCE} + P_{EXPANDER} = 0.7 + 1 = 1.7W$$

Celý TimePanel obsahuje dohromady dvanáct TimePanelu_1 (1x TA40-11SRWA a 1x MCP23016). Celkový maximální odebíraný výkon všech TimePanelu_1 pro zobrazení názvu lokality v celém zařízení je zhruba:

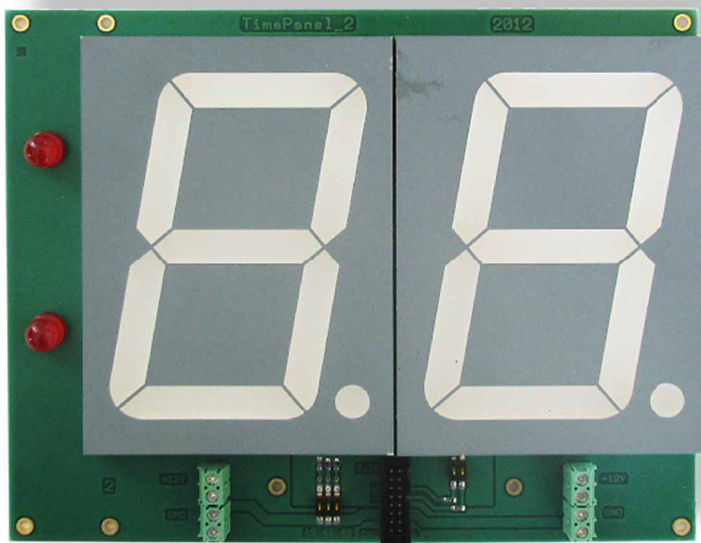
$$P_{TimePanel_1_ALL} = 12 \cdot P_{TimePanel_1} = 1.7 \cdot 12 = 20.4W \cong 21W$$

Jednotlivé výpočty předřadných odporů a ztrátových výkonů pro napájecí napětí 5V i 12V byly ověřeny pomocí simulačního programu NI MultiSim. Schéma zapojení, plošné spoje, seznam součástek a fotografie modulu TimePanel_1 jsou v příloze (Příloha II).

Protože modul TimePanel_1 s maticovým displejem pracuje v dynamickém režimu spínání (multiplex) sloupců je nutné předřadný odpor R_{LED} snížit zhruba na jednu pětinu původní hodnoty $R_{LED} = 33\Omega$. Při testování expandérů byl také změřen jeho skutečný odebíraný výkon $P_{EXPANDER} = 0.1W$, při nastavení a zapojení shodném jako v TimePanelu. A proto bude skutečný maximální odebíraný výkon modulu TimePanelu_1 mnohem nižší.

4.2.5 Modul TimePanel_2

TimePanel 2 slouží k zobrazení světového času (Hodiny, Minuty, Sekundy-2 LED diody) v daných lokalitách. Deska obsahuje jeden I2C expandér MCP23016 pro **dva sedmi segmentové LED displeje SA40-19SRWA**. Celkově TimePanel obsahuje čtyři desky TimePanel_2. Rozměry jedné desky TimePanelu_2 jsou **205x160 mm** (Obr. 33). Rozměry, zapojení a technické parametry sedmi segmentového displeje SA40-19SRWA a LED diody KINGBRIGHT - L-813ID jsou v příloze (Příloha III).



Obr. 33 Modul TimePanel_2 (TOP)

4.2.5.1 Výpočty předřadného odporu a ztrátových výkonu pro LED diody

$$U_{NAPAJECI} = 12V$$

$$V_{F(typ)} = 2V$$

$$I_{F(typ)} = 20mA$$

$$V_{CE(sat)} = 0.7V \quad - \text{viz graf } V_{CE(sat)} = f(I_{OUT})$$

$$U_{RLED} = U_{NAPAJECI} - V_{F(typ)} - V_{F(typ)} - V_{CE(sat)} = 12 - 2 - 2 - 0.7 = 7.3V$$

$$P_{LED} = 2 \cdot V_{F(typ)} \cdot I_{F(typ)} = 2 \cdot 2 \cdot 0.02 = 80mW$$

Předřadný odpor R_{LED} :

$$R_{LED} = \frac{U_{RLED}}{I_{F(typ)}} = \frac{7.3}{0.02} = 365\Omega$$

$$P_{RLED} = U_{RLED} \cdot I_{F(typ)} = 7.3 \cdot 0.02 = 146mW$$

Výkon potřebný pro LED diody s $R_{LED} = 365\Omega$:

$$P_{SEGMENT} = P_{RLED} + P_{LED} = 146 + 80 = 226mW \cong 230mW$$

Na TimePanelu jsou LED diody pouze na dvou modulech a celkový výkon je:

$$P_{TimePanel_2_LED_DIODY} = 2 \cdot P_{LED_DIODY} = 230mW \cdot 2 = 460mW$$

4.2.5.2 Výpočty předřadného odporu a ztrátových výkonů segmentů

$$U_{NAPAJECI} = 12V$$

$$V_{F(typ)} = 7.4V$$

$$I_{F(typ)} = 20mA$$

$$V_{CE(sat)} = 0.7V \quad - \text{viz graf } V_{CE(sat)} = f(I_{OUT})$$

$$U_{RLED} = U_{NAPAJECI} - V_{F(typ)} - V_{CE(sat)} = 12 - 7.4 - 0.7 = 3.9V$$

$$P_{LED} = V_{F(typ)} \cdot I_{F(typ)} = 7.4 \cdot 0.02 = 148mW$$

Předřadný odpor R_{LED} :

$$R_{LED} = \frac{U_{RLED}}{I_{F(typ)}} = \frac{3.9}{0.02} = 195\Omega$$

$$P_{RLED} = U_{RLED} \cdot I_{F(typ)} = 3.9 \cdot 0.02 = 78mW$$

Výkon potřebný pro jeden segment s $R_{LED} = 195\Omega$:

$$P_{SEGMENT} = P_{RLED} + P_{LED} = 148 + 78 = 226mW$$

Z důvodu zvýšení svítivosti displeje byl zvolen odpor $R_{LED} = 160\Omega$. Proto je nutno ztrátové výkony přepočíst.

$$I_{F(typ)} = \frac{U_{RLED}}{R_{LED}} = \frac{3.9}{160} = 24.4mA$$

$$P_{RLED} = U_{RLED} \cdot I_{F(typ)} = 3.9 \cdot 0.0244 = 95.2mW$$

$$P_{LED} = V_{F(typ)} \cdot I_{F(typ)} = 7.4 \cdot 0.0244 = 180.6mW$$

Výkon potřebný pro jeden segment s $R_{LED} = 160\Omega$:

$$P_{SEGMENT} = P_{RLED} + P_{LED} = 180.6 + 95.2 = 275.6mW \cong 280mW$$

Jelikož sedmi segmentový displej obsahuje 7 segmentů tak $P_{SEGMENT}$ je nutno vynásobit sedmi.

$$P_{LED_DISPLEJ} = P_{SEGMENT} \cdot 7 = 280mW \cdot 7 = 2W \cong 2.1W$$

Výkon $P_{LED_DISPLEJ}$ je maximální výkon, který může displej odebírat ze zdroje, protože je spočtený pro stav kdy budou svítit všechny segmenty displeje. Ve skutečnosti však najednou nikdy všechny segmenty displeje svítit nebudou, a proto bude odebíraný výkon nižší.

Modul TimePanel_2 obsahuje dva sedmi segmentové displeje SA40-19SRWA a jeden expandér MCP23016, který má maximální ztrátový výkon $P_{EXPANDER} = 1W$ (300mA, 3.3V). A proto maximální odebíraný výkon, který odebírá TimePanel_2 je:

$$P_{TimePanel_2} = 2 \cdot P_{LED_DISPLEJ} + P_{EXPANDER} = 2.1W \cdot 2 + 1 = 5.2W$$

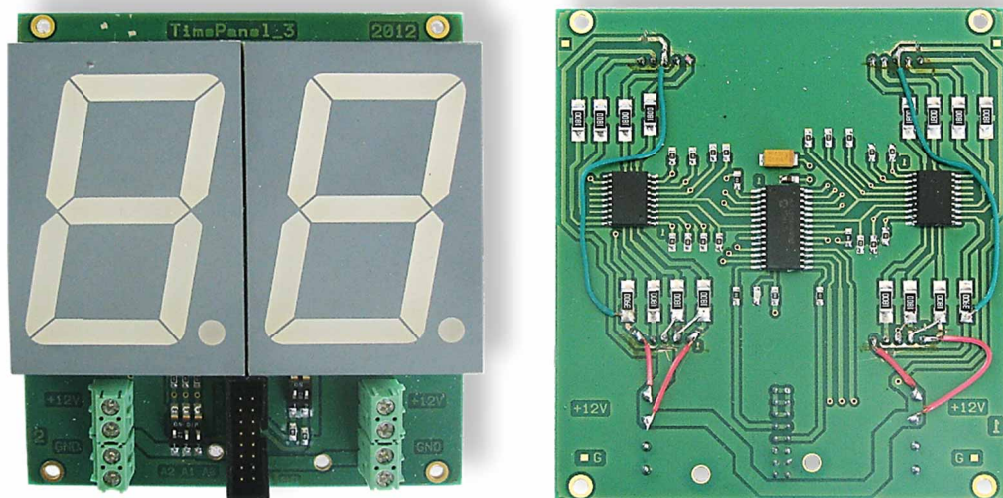
Celý TimePanel obsahuje dohromady čtyři TimePanel_2 (2x SA40-19SRWA a 1x MCP23016). Celkový maximální odebíraný výkon všech TimePanelu_2 pro zobrazení času v celém zařízení je zhruba:

$$P_{TimePanel_2_ALL} = 4 \cdot P_{TimePanel_2} + P_{TimePanel_2_LED_DIODY} = 5.2W \cdot 4 + 462mW = 21.3W \cong 22W$$

Jednotlivé výpočty předřadných odporů a ztrátových výkonů pro napájecí napětí 12V byly ověřeny pomocí simulačního programu NI MultiSim. Schéma zapojení, plošné spoje, seznam součástek a fotografie modulu TimePanel_2 jsou v příloze (Příloha III). Při testování expandérů byl také změřen jeho skutečný odebíraný výkon $P_{EXPANDER} = 0.1W$, při nastavení a zapojení shodném jako v TimePanelu. A proto bude skutečný maximální odebíraný výkon modulů TimePanelu_2 mnohem nižší.

4.2.6 Modul TimePanel_3

Modul TimePanel 3 slouží k zobrazení data (Den, Měsíc, Rok). Deska obsahuje jeden I2C expandér MCP23016 pro dva sedmi segmentové LED displeje SA23-12SRWA. Celkově TimePanel obsahuje šest desek TimePanel_3. Rozměry jedné desky TimePanelu_3 jsou **95.5x106.5 mm** (Obr. 34). Rozměry, zapojení a technické parametry sedmi segmentového displeje SA23-12SRWA jsou v příloze (Příloha IV).



Obr. 34 Modul TimePanel_3 (TOP a BOTTOM)

4.2.6.1 Výpočty předřadného odporu a ztrátových výkonu segmentu (DP)

$$U_{NAPAJECI} = 12V$$

$$V_{F(typ)} = 7.4V$$

$$V_{F_DP(typ)} = 3.7V$$

$$I_{F(typ)} = 20mA$$

$$V_{CE(sat)} = 0.7V \quad - \text{viz graf } V_{CE(sat)} = f(I_{OUT})$$

$$U_{RLED} = U_{NAPAJECI} - V_{F(typ)} - V_{CE(sat)} = 12 - 7.4 - 0.7 = 3.9V$$

$$P_{LED} = V_{F(typ)} \cdot I_{F(typ)} = 7.4 \cdot 0.02 = 148mW$$

$$U_{RLED_DP} = U_{NAPAJECI} - V_{F_DP(typ)} - V_{CE(sat)} = 12 - 3.7 - 0.7 = 7.6V$$

$$P_{LED_DP} = V_{F(typ)} \cdot I_{F(typ)} = 3.7 \cdot 0.02 = 74mW$$

Předřadný odpor R_{LED} :

$$R_{LED} = \frac{U_{RLED}}{I_{F(typ)}} = \frac{3.9}{0.02} = 195\Omega$$

$$P_{RLED} = U_{RLED} \cdot I_{F(typ)} = 3.9 \cdot 0.02 = 78mW$$

Předřadný odpor R_{LED_DP} :

$$R_{LED_DP} = \frac{U_{RLED_DP}}{I_{F(typ)}} = \frac{7.6}{0.02} = 380\Omega$$

$$P_{RLED_DP} = U_{RLED_DP} \cdot I_{F(typ)} = 7.6 \cdot 0.02 = 152mW$$

Výkon potřebný pro jeden segment s $R_{LED} = 195\Omega$:

$$P_{SEGMENT} = P_{RLED} + P_{LED} = 148 + 78 = 226mW$$

Výkon potřebný pro DP s $R_{LED_DP} = 380\Omega$:

$$P_{DP} = P_{RLED_DP} + P_{LED_DP} = 152 + 74 = 226mW$$

Z odporové řady byl vybrán odpor $R_{LED} = 220\Omega$ a $R_{LED_DP} = 390\Omega$. Proto je nutno ztrátové výkony přepočíst. $I_{F(typ)} = \frac{U_{RLED}}{R_{LED}} = \frac{3.9}{220} = 17.7mA$

$$P_{RLED} = U_{RLED} \cdot I_{F(typ)} = 3.9 \cdot 0.0177 = 69.3mW$$

$$P_{LED} = V_{F(typ)} \cdot I_{F(typ)} = 7.4 \cdot 0.0177 = 131.1mW$$

Výkon potřebný pro jeden segment s $R_{LED} = 220\Omega$:

$$P_{SEGMENT} = P_{RLED} + P_{LED} = 131.1 + 69.3 = 200.5mW \cong 201mW$$

Pro desetinnou tečku DP s předřadným odporem $R_{LED_DP} = 390\Omega$:

$$I_{F(typ)} = \frac{U_{RLED_DP}}{R_{LED_DP}} = \frac{7.6}{390} = 19.5mA$$

$$P_{RLED_DP} = U_{RLED_DP} \cdot I_{F(typ)} = 7.6 \cdot 0.0195 = 148.2mW$$

$$P_{LED_DP} = V_{F(typ)}_{DP} \cdot I_{F(typ)} = 3.7 \cdot 0.0195 = 73mW$$

Výkon potřebný pro desetinnou tečku DP s $R_{LED_DP} = 390\Omega$:

$$P_{DP} = P_{RLED_DP} + P_{LED_DP} = 149 + 73 = 222mW \cong 230mW$$

Jelikož sedmi segmentový displej obsahuje sedm segmentů tak $P_{SEGMENT}$ je nutno vynásobit sedmi a přičíst k němu výkon desetinné tečky P_{DP} .

$$P_{LED_DISPLEJ} = P_{SEGMENT} \cdot 7 + P_{DP} = 201mW \cdot 7 + 230mW = 1.637W \cong 1.7W$$

Výkon $P_{LED_DISPLEJ}$ je maximální výkon který může displej odebírat ze zdroje, protože je spočtený pro stav kdy budou svítit všechny segmenty displeje a DP. Ve skutečnosti však najednou nikdy všechny segmenty displeje svítit nebudou a proto bude odebíraný výkon nižší.

Modul TimePanel_3 obsahuje dva sedmi segmentové displeje SA23-12SRWA a jeden expandér MCP23016, který má maximální ztrátový výkon $P_{EXPANDER} = 1W$ (300mA, 3.3V). Proto maximální odebíraný výkon, který odebírá TimePanel_3 je:

$$P_{TimePanel_3} = 2 \cdot P_{LED_DISPLEJ} + P_{EXPANDER} = 1.7W \cdot 2 + 1 = 4.4W$$

Celý TimePanel obsahuje dohromady šest TimePanelu_3 (2x SA23-12SRWA a 1x MCP23016). Celkový maximální odebíraný výkon všech TimePanelu_3 pro zobrazení data v celém zařízení je zhruba:

$$P_{TimePanel_3_{ALL}} = 6 \cdot P_{TimePanel_3} = 4.4W \cdot 6 \cong 26.4W$$

Jednotlivé výpočty předřadných odporů a ztrátových výkonů pro napájecí napětí 12V byly ověřeny pomocí simulačního programu NI MultiSim. Schéma zapojení, plošné spoje, seznam součástek a fotografie modulu TimePanel_3 jsou v příloze (Příloha IV). Při testování expandérů byl změřen jeho skutečný odebíraný výkon $P_{EXPANDER} = 0.1W$, při nastavení a zapojení shodném jako v TimePanelu. A proto bude skutečný maximální odebíraný výkon modulů TimePanelu_3 mnohem nižší.

4.2.7 Rozšiřující deska ‚Expansion board‘

Hlavním úkolem rozšiřující desky ‚Expansion board‘ je rozvod **datových** (*I2C1, I2C0*) a **napájecích signálů** (*3.3V, GND*) po TimePanelu, především pro expandéry na jednotlivých modulech TimePanel 1 až 3 (Obr. 36). Tyto datové a napájecí signály jsou vyvedeny na čtyři výstupní konektory MLW16G, ze kterých jsou pak pomocí 16- žilových **plochých kabelů** rozvedeny na jednotlivé moduly TimePanelu.

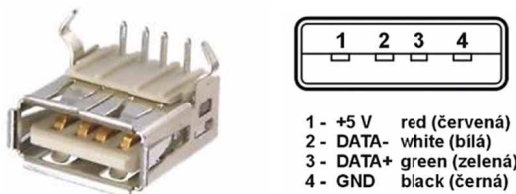
Konektory MLW16G na desce:

- **J7 a J8** – output – pro TimePanel 2 a 3 – 3.3V , GND , I2C0 (*SDA, SCL*)
- **J9a J10** – output - pro TimePanel 1 – 3.3V, GND , I2C1 (*SDA, SCL*)

Datové signály sběrnice I2C0 a I2C1 jsou s TWR-PXN20 (Tabulka 2) připojeny na desku na konektory J12 a J13 . Konektory J11(I2C0) a J14(I2C1) slouží jako sondy pro připojení osciloskopu a sledování signálů sběrnice.

- **J12** – input – z TWR-PXN20 - sběrnice I2C0 (pin 1 – *SDA*, pin 2 – *SCL*)
- **J13** – input – z TWR-PXN20 – sběrnice I2C1 (pin 1 – *SDA*, pin 2 – *SCL*)

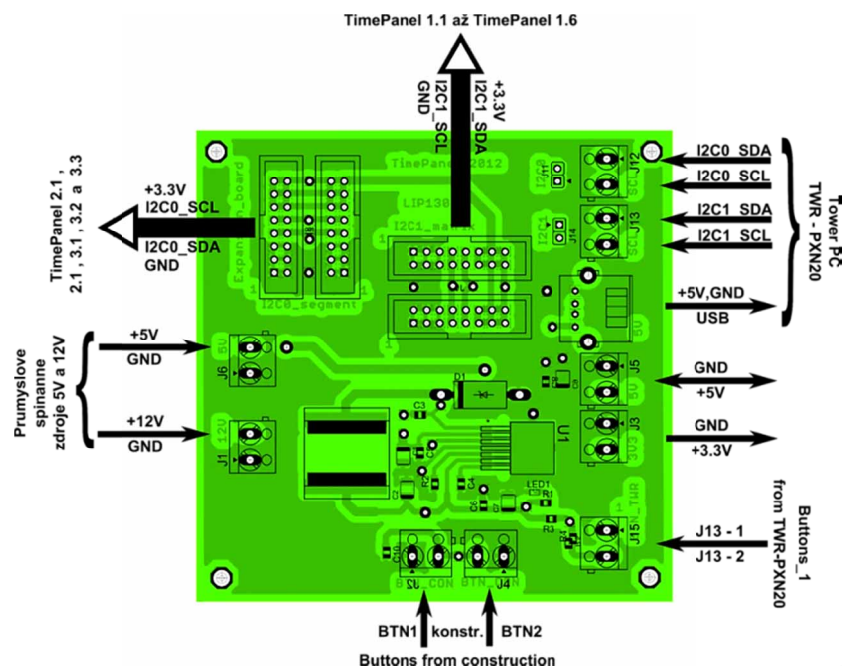
Na rozšiřující desce bylo nutné pro napájení expandérů navrhnout spínaný zdroj s pevným výstupním napětím 3.3V (viz Kapitola 5.2.7.1). Rozšiřující deska dále slouží k napájení řídicí jednotky TWR-PXN20 pomocí 5V. Toto napětí je na desku přivedeno z průmyslového spínaného 5V zdroje na vstupní konektor J6, ze kterého je vyvedeno na výstupní konektor J5 (*ARK120-2*) a výstupní konektor USB. Na obrázku (Obr. 35) je použitý konektor USB typu A – host a jeho zapojení.



Obr. 35 USB konektor typu A a jeho zapojení

Rozšiřující deska je také použita k rozšíření tlačítka BUTTON_1 s Power-PC (TWR-PXN20) o dva tlačítka v konstrukci TimePanelu . Tlačítko BUTTON_1 je přivedeno na konektor J15 s TWR-PXN20 z konektoru J13 – pinu 1 a 2. Z ‚Expansion Board‘ je pak tlačítko BTN_1 vyvedeno na externí tlačítka v konstrukci z konektorů J2 a J4. Externí tlačítka J2 a J4 jsou paralelně spojena na DPS. V případě že bude potřeba mít funkční jak externí tlačítka v konstrukci tak tlačítko BTN1 na desce TWR-PXN20 , je třeba osadit odpor R5 na rozšiřující desku. Pokud nebude odpor R5 osazen, budou funkční jen externí tlačítka v konstrukci.

- **J15** – tlačítko BTN1 s TWR-PXN20 – konektor J13 – pin 1 a 2
- **J2 a J4** – externí tlačítka umístěné v konstrukci



Obr. 36 Vstupy a výstupy z rozšiřující desky pro TimePanel

Celkové schéma zapojení, seznam použitých součástek a plošné spoje s rozměry rozšiřující desky jsou v příloze (Příloha V).

4.2.7.1 Návrh spínaného zdroje pro pevné výstupní napětí 3.3V

TimePanel obsahuje dva průmyslové spínané napájecí zdroje 12V a 5V pro napájení modulu TimePanel 1 až 3, ale jelikož expandéry na jednotlivých modulech vyžadují napájení 3.3V, je nutné navrhnout zdroj s výstupním napětím 3.3V.

Při návrhu bylo bráno v úvahu především výkonové zatížení zdroje. To závisí zejména na spotřebě expandérů. Výrobce uvádí maximálně 300mA při 3.3V pro jeden expandér a jelikož je v TimePanelu použito 22 kusů expandérů, je nutné, aby zdroj byl schopen dodávat okolo 6.6A při 3.3V (zhruba 22W). Tato uváděná spotřeba expandérů je velmi vysoká a proto bylo provedeno orientační měření odebíraného proudu expandérem (viz Kapitola 6.1). Z tohoto měření bylo zjištěno, že odebíraný proud expandérem je zhruba 20mA, to znamená, že skutečný odebíraný proud všech expandérů v TimePanelu je okolo 0.5A (zhruba 2W).

Při návrhu zdroje bylo možné zvolit lineární nebo spínaný zdroj. Oba tyto typy zdrojů mají svoje výhody i nevýhody. Pro tuto aplikaci byl vybrán jako vhodnější spínaný zdroj, jelikož má vyšší účinnost a nižší ztráty než lineární. Posléze byl vybrán DC/DC měnič LM2673S-3.3, který má účinnost okolo 94%, maximální výstupní proud 3A a spínací frekvenci 260kHz. Návrh externích součástek a popis regulátoru LM2673S-3.3 je popsán v příloze (Příloha V).

Provozní podmínky regulátoru LM2673S-3.3 v aplikaci TimePanel :

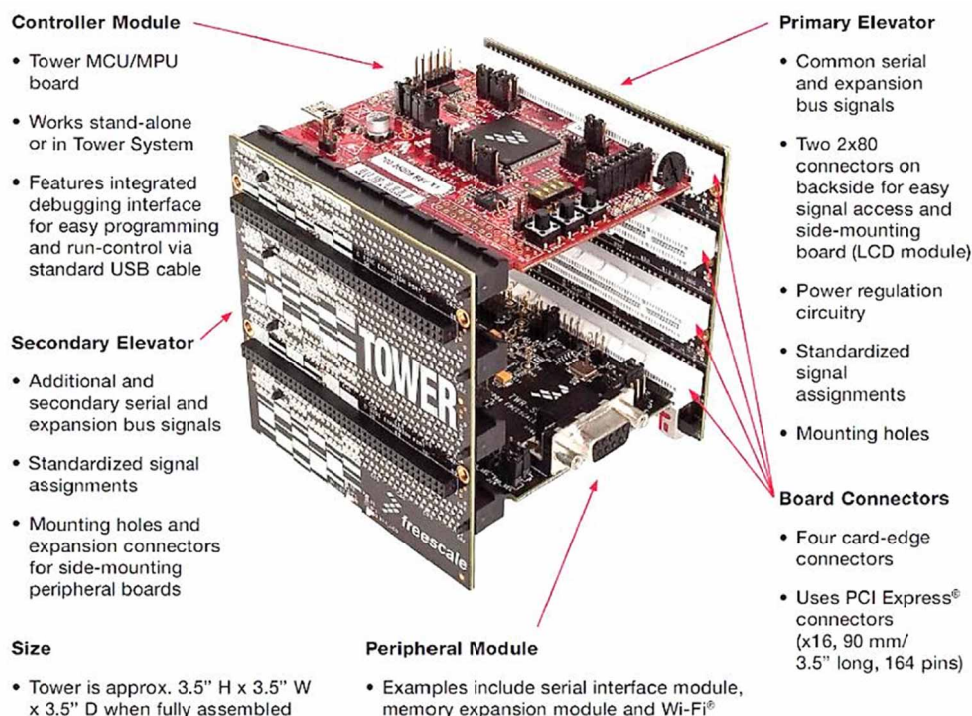
- Vstupní napětí: $U_{IN}=12V$
- Výstupní napětí: $U_{OUT}=3.3V$
- Výstupní proud: $I_{OUT}=3A$ - maximální hodnota

4.3 Výběr hardwarových součástí TimePanelu

V této podkapitole jsou stručně popsány použité hardwarové součásti v TimePanelu jako napájecí zdroje a řídicí jednotka.

4.3.1 Řídicí jednotka – Freescale Power PC - modul PXN20

Freescale Tower System je modulární vývojová platforma pro 8-mi, 16-ti a 32-bitové mikrokontroléry a mikroprocesory pomocí metody rychlého návrhu (*Rapid prototyping*). Základem systému jsou dvě desky výtahů (*Elevators*) se standardními PCI konektory propojeny jedna ke druhé. Levý výtah má navíc zdroj napětí 5V a 3.3V. Mezi dva výtahy je možné vložit desku s mikroprocesorem a další 3 rozšiřující desky. Desky je možné vložit v libovolném pořadí. Napájení je možné z USB konektory (*Mini USB*) nebo šroubovací svorky s desky výtahu nebo ze základní procesorové desky (Obr. 37).



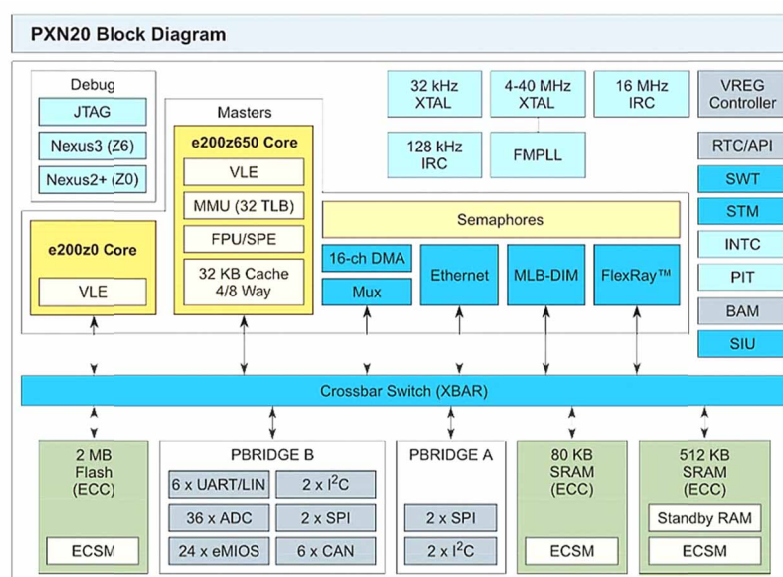
Obr. 37 Freescale Tower System [8]

Mezi základní (procesorové desky) patří osmi bitové desky s mikrokontroléry HCS08, šestnácti bitové desky s mikrokontroléry HCS12 a DSC a největší skupinou jsou desky s 32-bitovými mikroprocesory řad Kinetis, Power PC a Vybrid.

Mezi rozšiřující desky pak patří sériová deska s Ethernetem, sběrnici CAN, RS485 dále pak paměťový modul, analogový modul, modul pro řízení motorů nebo *Wi-Fi* modul. V případě potřeby existuje prototypová deska s pájecím polem pro vytvoření libovolného zákaznického modulu. Všechny moduly patřící do rodiny Freescale Tower System mají podporu RT operačního systému MQX popř. operačního systému Linux.[7][13]

4.3.1.1 Procesor PXN20

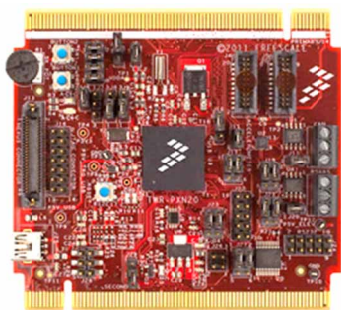
Procesor PXN20 je kompatibilní 32-bit mikroprocesor postaven na Power Architecture® technologii. Tyto PXN20 jsou navrženy pro potřeby jednočipových průmyslové aplikací, síťových aplikací a jsou navrženy tak, aby zůstal poměr vysokého výkonu a velké paměti při zachování nízké spotřeby energie. Jejich jádro a sběrnice architektura nabízí vysoko výkonnou a optimalizovanou správu, zpracování a výměnu dat mezi různými typy komunikačních protokolů. To vše se odvíjí od rozvoje infrastruktury stávající Power Architektury® a PXN20 je podporováno spoustou softwarových ovladačů a operačním systémem, který má pomoci při uživatelské implementaci SW (Obr. 38). [8][15]



Obr. 38 Blokový diagram procesoru PXN20 [8]

4.3.1.2 Hlavní modul TWR- PXN20

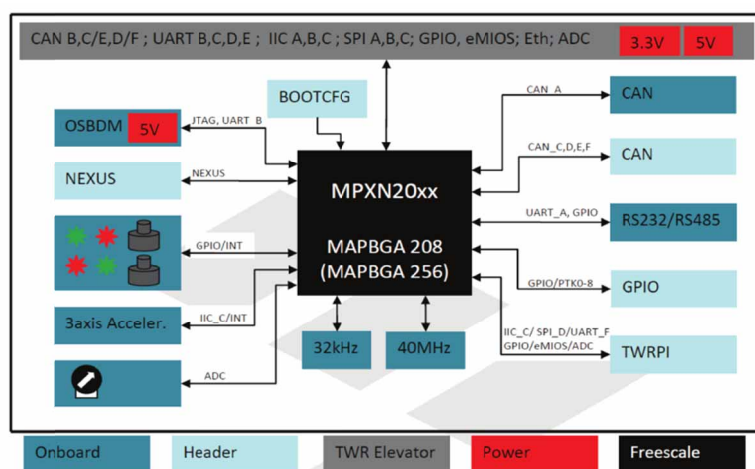
TWR-PXN20 je modulární vývoj platforma, která umožňuje *rapid prototyping* a nástroj *re-use* pomocí překonfigurování hardware. Mikroprocesory PXN20 32-bit dual core Power Architecture® podporují celou řadu komunikačních protokolů, což umožňuje efektivní a spolehlivé průmyslové systémy se špičkovým výkonem (Obr. 39).[8] [15]



Obr. 39 TWR- PXN20 [7]

4.3.1.2.1 Vlastnosti modulu TWR- PXN20

- MPXN2020VMG116 MCU (až 116 MHz 32-bit dual core MCU: e200z6 a e200z0, 2 MB on-chip Flash, 592 KB on-chip SRAM, 32-kanálový DMA, 6x UART, 3x SPI, 6x CAN, 4x I²C, 36-kanálový 10-bit ADC, rychlý Ethernet controller (FEC), Debug JTAG rozhraní a Nexus 3, 16-bit, interní časovače 8-kanálové, 32 -bit, tepl. rozsah -40 ° C / +105° C, pouzdro čipu 208 MAPBGA, režim nízké spotřeby)
- CAN, RS485 a RS232
- Čtyři LED diody a dva mechanická tlačítka pro uživatelské rozhraní
- 3-osý akcelerometr (MMA8451Q)
- Univerzální TWRPI zásuvka (Tower plug-in modul)
- OSJTAG open source JTAG ladicí rozhraní (MC9S08JM60)
- OSJTAG virtuální pomoci USB



Obr. 40 Blokové schéma TWR-PXN20 [8]

4.3.1.3 Sériový modul TWR - SER

Tento modul je periferní deskou a poskytuje systému Freescale Tower přídavná komunikační rozhraní. Periferní modul je navržen tak, aby jej bylo možné kombinovat a používat s jinými MCU moduly a periferními moduly v rámci Freescale Tower System. Tato deska obsahuje následující (Obr. 41). [14]



Obr. 41 TWR-SER [14]

- Ethernet konektor RJ45 s integrovanými oddělovacími transformátory a LED diodami poskytující fyzickou vrstvu pro rozhraní Ethernet.
- Vysokorychlostní fyzické rozhraní pro USB.
- USB Host, Device, OTG a s Mini-AB konektorem
- RS232 , RS485 a CAN transceiver

4.3.2 Napájecí zdroje

V původním studentském návrhu hardwarových součástí byl pro napájení celého TimePanelu zvolen průmyslový spínaný zdroj s výstupním napětím 24V. Úbytek napětí na použitých LED displejích je u sedmi segmentu 7V a u matic 2.4V. To znamená, že při napájení pomocí 24V by byl úbytek na předřadných odporech zhruba 17V (21.6V), což představuje velkou výkonovou ztrátu. Z těchto důvodů je použití tohoto návrhu a 24V zdroje nevhodné.

V současném návrhu řešení, měl TimePanel obsahovat pouze jeden 12V spínaný napájecí zdroj pro napájení všech modulů, ale jelikož by **stále** byly na předřadných odporech maticových displejů velké ztrátové výkony, byl tento návrh v průběhu pozměněn a proto TimePanel obsahuje dva průmyslové napájecí zdroje (12V a 5V).

4.3.2.1 Spínaný zdroj 12V

Pro napájení rozšiřující desky a modulů TimePanelu 2 a 3 je vybrán průmyslový spínaný zdroj s krytem XP POWER PSU 100W 12V (Obr. 42).



Výkon zdroje	100W
Vstupní napětí	85V až 264V
Výstupní napětí	12V
Výstupní proud	8.33A
Hmotnost	459g
Rozměry	42x 159x 93 mm

Obr. 42 Zdroj XP POWER 12Va jeho parametry

4.3.2.2 Spínaný zdroj 5V

Pro napájení modulů TimePanelu 1 je zvolen průmyslový spínaný zdroj s krytem TDK LAMBDA PSU LS50-5 50W 5A (Obr. 43).



Výkon zdroje	50W
Vstupní napětí	88V až 264V
Výstupní napětí	5V
Výstupní proud	10A
Hmotnost	350g
Rozměry	97x 36x 99 mm

Obr. 43 Zdroj LS50-5 a jeho parametry

4.4 Softwarová - realizační část

V této kapitole zabývající se realizací softwaru, jsou popsány nejpoužívanější příkazy pro příkazový řádek, režimy přepínání lokalit u TimePanelu a na závěr kompletní aplikace pro zařízení pomocí UML diagramu.

V původním studentském návrhu SW pomocí UML byly vytvořeny jen tři základní UML diagramy popisující aplikaci s nevhodnou strukturou, třídami a propojením. Navržené UML diagramy nesplňovaly požadavky pro TimePanel, nebyla řešena synchronizace času ani posílání dat po I2C a především byly vyvíjeny pro aplikaci TimePanel s BareMetal bez použití OS. Z těchto důvodů, bylo rozhodnuto, že bude pro TimePanel vyvinuta nová (vlastní) aplikace s využitím RTOS, která bude průběžně testována a po odladění budou vytvořeny UML diagramy.

4.4.1 Příkazy a nastavení periferii v MQX

V případě, že je nutno použít libovolnou periferii (rozhraní) na cílovém zařízení TWR-PXN20 s MQX, je nezbytné ji softwarově povolit či zakázat. K povolení nebo zakázání jednotlivých rozhraní procesoru slouží soubor `user_config.h`. Pro aplikaci TimePanel je nutno v tomto souboru povolit vstupně/výstupní porty TTYA a TTYB, I²C sběrnice I2C0 a I2C1, RTCs a také SHELL.

Po povolení vstupně/výstupních portů TTYA, TTYB je možné pomocí příkazového řádku SHELL komunikovat se zařízením přes terminál. V terminálu je možné sledovat chod aplikace, nastavit sběrnice I2C, síťové připojení pro synchronizaci času a manuálně přenastavovat čas, datum a režimy přepínání. Jelikož už zařízení využívá připojení k síti přes Ethernet, je navíc v aplikaci vytvořen Telnet Server využívající TCP/IP protokol, který uživateli poskytuje síťový virtuální terminál pro komunikaci se zařízením. S TimePanelem tedy lze komunikovat přes terminál pomocí sériové linky (USB) nebo sítě (Telnet Server). Ke komunikaci byl použit terminál Hercules SETUP utility od společnosti HW-Group.

Cesta k souboru `user_config.h` :

`C:\Program Files\Freescale\Freescale MQX 3.8\config\twrpxn20\user_config.h`

V operačním systémech MQX je standardně nastavena v BSP maximální rychlost přepínání úloh na 5ms (*task_tick*) pomocí proměnné `BSP_ALARM_FREQUENCY (200)`. Z tohoto důvodu je například u funkce časového zpoždění `_time_delay(hodnota)` minimální hodnota zpoždění 5ms. Jelikož v aplikaci TimePanel je zapotřebí časového zpoždění 1ms, je nutné změnit maximální rychlost přepínání úloh na 1ms. To lze provést změnou hodnoty proměnné `BSP_ALARM_FREQUENCY` z 200Hz(5ms) na 1000Hz(1ms). Na obrázku (Obr. 44) je znázorněna definice a cesta k hlavičkovému souboru této proměnné.


```
// C:\Freescale\Freescale MQX 3.8\lib\twrpxn20.cw10\bsp\twrpxn20.h
/*      TIMER DRIVER CONFIGURATION      */
/* MGCT: <category name="BSP Hardware Options"> */
/* The clock tick rate in miliseconds - be cautious to keep this value such
** that it divides 1000 well.
** MGCT: <option type="number" min="1" max="1000"/> */

#ifndef BSP_ALARM_FREQUENCY
#define BSP_ALARM_FREQUENCY      (200)
#endif
```

Obr. 44 Definice proměnné *BSP_ALARM_FREQUENCY*

Tuto změnu nelze provést v knihovně BSP, ale proměnnou *BSP_ALARM_FREQUENCY* je nutné nadefinovat v *,user_config.h‘* a tam novou hodnotu proměnné nastavit. V podkapitolách (Kapitoly 5.3.3) jsou průběžně v textu uvedeny (popsány) vlastnosti a parametry vlastních příkazů a funkcí, využívaných v příkazovém řádku nebo důležitých pro správný chod aplikace TimePanel. Vlastní příkazy byly vytvořeny pro účely aplikace.

4.4.2 Přepínání lokalit TimePanelu

Jelikož TimePanel slouží k zobrazení světového času v šesti různých městech-lokalitách na světě, je zapotřebí nějakým způsobem zajistit přepínání lokalit zobrazovaných na TimePanelu . Pro přepínání režimu a lokalit je k dispozici:

- tlačítko *,Button_1‘* na desce TWR-PXN20
- vyvedení GPIO nebo IRQ na univerzální konektor J22 nebo elevátorové desky

U TimePanelu si můžeme zvolit režim přepínání automatický (AUTO) nebo manuální (MAN). U automatického režimu přepínání dochází na TimePanelu po uplynutí nastaveného časového intervalu k automatické změně zobrazované lokality, času a data pro danou lokalitu. U manuálního režimu dojde ke změně zobrazované lokality po stisknutí tlačítka, pokud nedojde ke stisku tlačítka je zobrazená stále stejná lokalita. Bylo navrženo mnoho variant, jakým způsobem lze přepínat jednotlivé lokality a režimy přepínání. V podkapitolách níže jsou popsány jednotlivé možné varianty a návrhy přepínání.

4.4.2.1 Pouze manuální režim přepínání

Při této variantě k přepnutí lokality z jedné na druhou dojde pouze v případě stisku tlačítka na zařízení. Tato varianta by měla uplatnění v situacích, kdy by bylo potřeba mít na panelu zobrazen čas jen pro jednu danou lokalitu, a přepínání lokalit by prováděla obsluha zařízení stiskem tlačítka. Pro tuto aplikaci není tato samostatná varianta vhodná.

4.4.2.2 Pouze automatický režim přepínání

Při této variantě by docházelo k automatickému přepínání lokalit po uplynutí zadaného časového intervalu. Časový interval jak dlouho by byla lokalita zobrazena, než dojde k přepnutí na jinou, by bylo možné nastavit pomocí příkazového řádku Shell. Tato možnost by se dala použít pouze v případě, že by bylo potřeba znát aktuální čas a datum ve všech šesti lokalitách. Při tomto režimu není využito tlačítko. Pro tuto aplikaci není čistě automatický režim zcela vhodný.

4.4.2.3 Automatický i manuálním režim - Varianta 1

Tento režim má vypadat pouze jako manuální režim s tím, že při stisku tlačítka dochází k přepnutí lokalit a do automatického režimu se lze dostat v mezi stavu mezi poslední a první lokalitou, pokud v tomto mezi stavu vyčkáme alespoň 10 sekund bez stisku tlačítka. V automatickém režimu se lokality přepínají co 10 sekund.

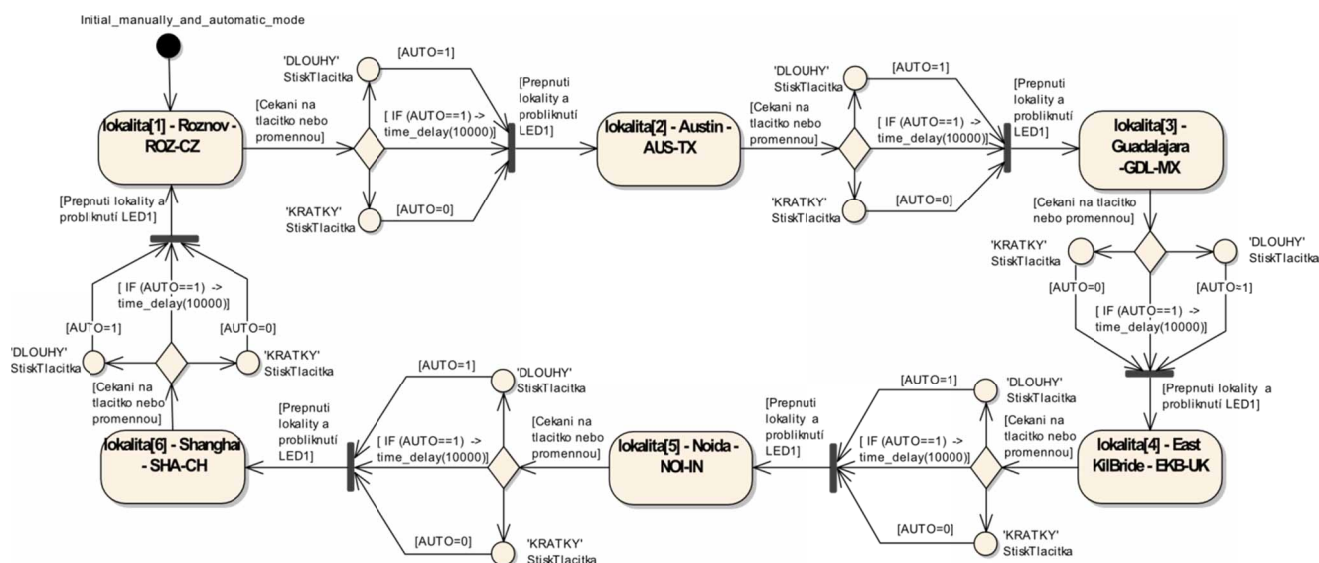
Z automatického režimu do manuálního se lze dostat stiskem tlačítka v libovolném okamžiku. Původně podle zadaných požadavků měla být tato varianta použita u TimePanelu, ale jelikož má mnoho nevýhod, byly navrženy jiné a vhodnější varianty pro TimePanel.

4.4.2.4 Automatický i manuálním režim - Varianta 2

Tato varianta spojuje dohromady jak automaticky tak manuální režim přepínání lokalit a je zapotřebí použít tlačítko. Při této variantě rozlišujeme dva různé stisky tlačítka:

- **Krátký** – netrvá déle jak 2 sekundy – stiskneme a hned pustíme tlačítko
- **Dlouhý** - delší jak 3 sekundy – stiskneme a držíme alespoň 3 sekundy

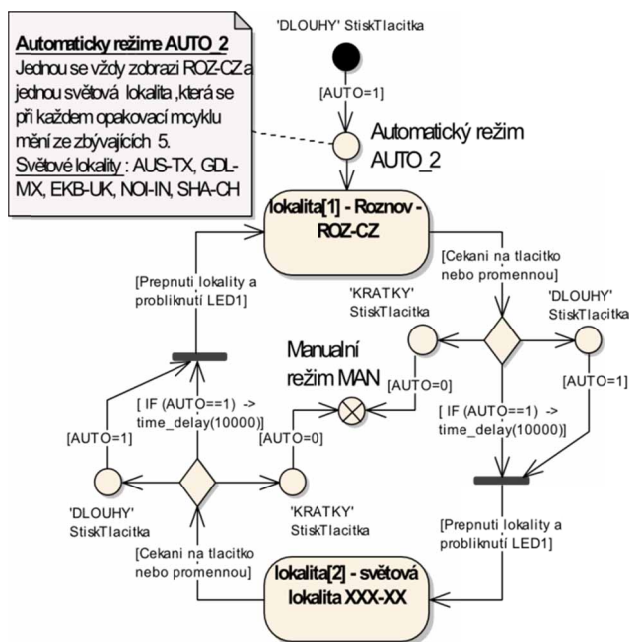
Při inicializaci SW je nastaven manuální režim přepínání a to znamená, že přepínáme mezi jednotlivými lokalitami na TimePanelu pomocí ‚krátkého‘ stisku tlačítka. Pokud chceme přejít do automatického režimu přepínání, provedeme ‚dlouhý‘ stisk tlačítka a dojde k přepnutí z MAN na AUTO režim. V automatickém režimu (*AUTO_1*) je všech šest lokalit postupně zobrazováno po sobě na TimePanelu. Z automatického režimu na manuální režim lze přejít ‚krátkým‘ stiskem tlačítka, kdy systém reaguje okamžitým přepnutím lokality a nastavením manuálního režimu. U této varianty je také navíc indikace přepnutí lokality pomocí LED diody (‚LED1‘), kdy při přepnutí lokality dojde k probliknutí LED1 na desce TWR-PXN20. Pomocí této indikace je možné zjistit, v jakém režimu se systém nachází (*AUTO- MAN*) nebo zda vůbec běží a ‚nezamrzl‘. Při inicializaci systému je nastaven manuální režim přepínání s výchozí lokalitou 1 – Rožnov pod Radhoštěm (Obr. 45) a viz příloha (Příloha VII).



Obr. 45 Přepínání lokalit – automaticky (*AUTO_1*) i manuální režim – varianta 2 - část varianty 3

4.4.2.5 Automatický i manuálním režim - Varianta 3

Tato varianta rozšiřuje variantu číslo dvě o další automatický režim přepínání lokalit (*AUTO_2*). Automatický režim přepínání lokalit *AUTO_2* nezobrazuje postupně po sobě všech šest lokalit jako režim *AUTO_1*, ale zobrazí vždy jednu Rožnov pod Radhoštěm (*ROZ-CZ*) a jednu světovou lokalitu, kdy se při každém cyklu světová lokalita mění ze zbývajících pěti. Při inicializaci systémů je nastaven automatický režim *AUTO_2* jako výchozí, ale při běhu aplikace je možné pomocí příkazového řádku *SHELL* a příkazu *switching_mode_set* nastavit automatický režim *AUTO_1* nebo *AUTO_2*.



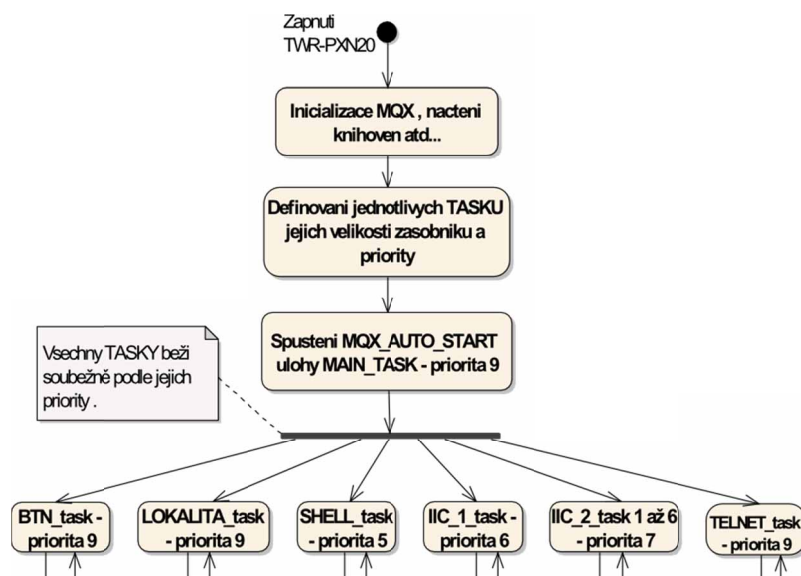
Obr. 46 Přepínání lokalit – automatický režim (*AUTO_2*) – část varianty 3 – finální verze

U této varianty je možné si zvolit ze tří různých režimů přepínání lokalit a to manuální režim *MAN*, automatický režim *AUTO_1* a *AUTO_2*. Stavový diagram této varianty je znázorněn na obrázcích (Obr. 45 a 46). Ze všech variant byla vybrána a implementována do systému tato varianta 3 (+2 varianta).

4.4.3 Aplikace pro TimePanel

Aplikace pro TimePanel běží na vestavěném systému s procesorem PXN20 pod operačním systémem reálného času FreeScale MQX . Aplikace je napsaná v programovacím jazyce ,C‘ . Tento software zajišťuje chod celého TimePanelu a je rozdělen do jednotlivých úloh, které řídí chod jednotlivých periférií systému (I2C,SNTP,GPIO...).

Při zapnutí TimePanelu s řídicí jednotkou dojde k inicializaci a načtení operačního systému MQX s aplikací pro zařízení. Po načtení potřebných knihoven a definování jednotlivých úloh s prioritami, dojde k automatickému spuštění startovací úlohy, Main Task‘, ve které jsou inicializovány a testovány jednotlivé periferie a moduly. Nakonec je v úloze ‚Main Task‘ vytvořeno a spuštěno dvanáct předem definovaných úloh s určenými prioritami (Obr. 47).

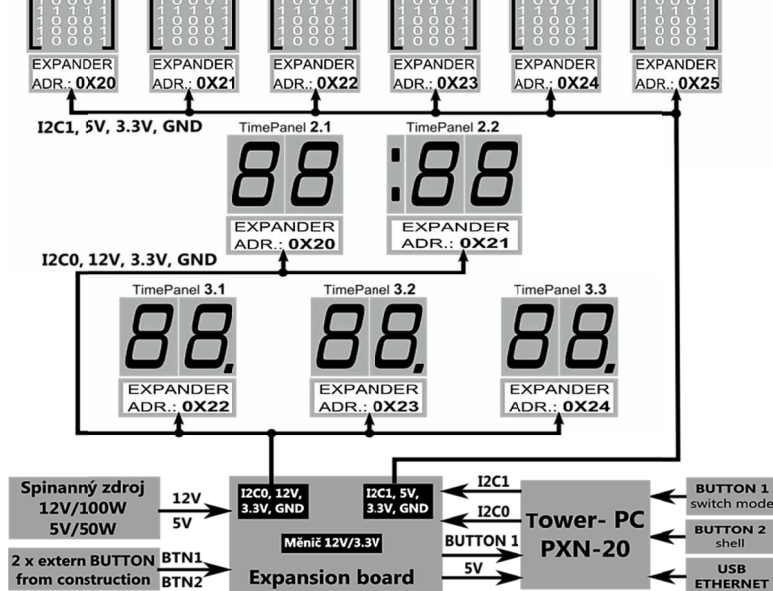


Obr. 47 Aplikace TimePanel - spuštění a jednotlivé úlohy

Jednotlivé úlohy běží v nekonečných smyčkách. Jejich priority byly nastaveny při spuštění systému, tak aby každá z úloh vykonala svůj úkol v daném čase. Každá z jednotlivých úloh aplikace řídí určitou periférii nebo vykonává stanovený úkol (Tabulka 4).

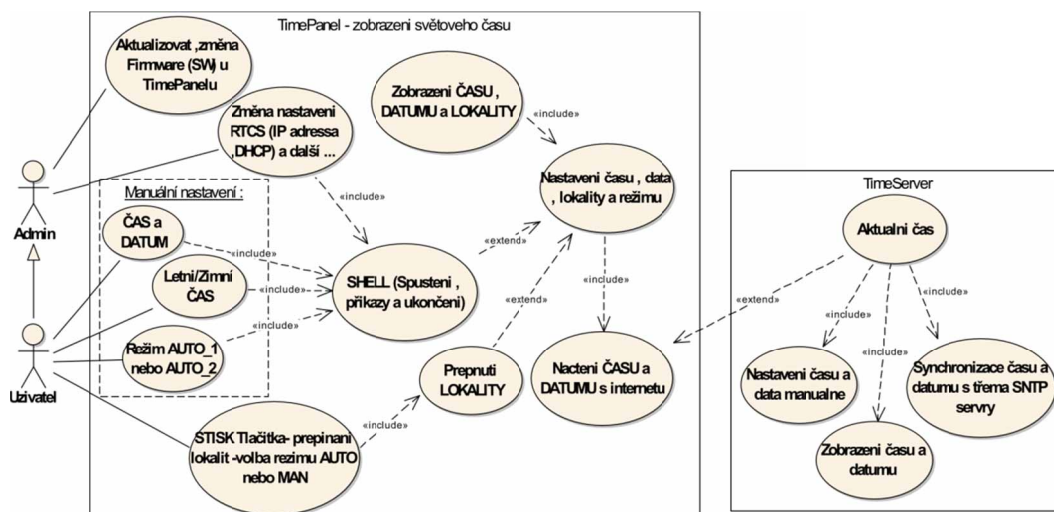
Úloha	Priorita	Úkol úlohy	Použité periferie
MAIN Task	9	Inicializace a testování periférií a modulu	TimePanel
BTN Task	9	Vyčítání hodnot pinu GPIO	Button 1, 2 a externí Button
LOKALITA Task	9	Přepínání jednotlivých lokalit a volba režimu přepínání (AUTO a MAN)	Button 1 a externí Button
SHELL Task	5	Komunikace a nastavení TimePanelu pomocí terminálu	USB (Ethernet) +BTN_2
IIC_1 Task	6	Zobrazování hodin a dat na sedmi segmentových displejích	I2C0
IIC_2 Task 1 až 6	7	Zobrazování názvu lokality na maticových displejích (IIC2_task_1až 6)	I2C1
TELNET_Task	9	Vytvoření Telnet serveru pro komunikaci se zařízením přes příkazový řádek s Ethernetem	Ethernet (TELNET)

Tabulka 4 Jednotlivé úlohy, jejich popis a priority

[illegible]

Obr. 48 Softwarové blokové schéma s adresami expandérů

Uživatelé, kteří budou s TimePanelem pracovat, jsou rozděleni do dvou skupin „Admin“ a „Uživatel“. Každá z těchto skupin má oprávnění jen pro dané úkoly viz obrázky (Obr. 49).



Obr. 49 Use case diagram – TimePanel

- **Admin** – členové této skupiny jsou oprávněni jednak pracovat s hodinami, stejně tak, jako běžní uživatelé, ale navíc jsou schopni aktualizovat a změnit Firmware, nastavit a změnit připojení k *Ethernetu* pomocí pevné IP adresy nebo DHCP a zvolit používané Time Servery pro synchronizaci času.

- **Uživatel** – v této skupině se nachází všichni běžní uživatelé, kteří jsou oprávněni k volbě režimu TimePanelu, přepínání zobrazovaných lokalit a manuálnímu nastavení hodin, dat a letního/zimního času pomocí SHELL (*TELNET*). Běžní uživatelé si rovněž mohou manuálně zvolit mezi automatickým režimem přepínání AUTO_1 nebo AUTO_2 , opět pomocí SHELL (*TELNET*). Jiné úkoly jsou této skupině nepřístupné.

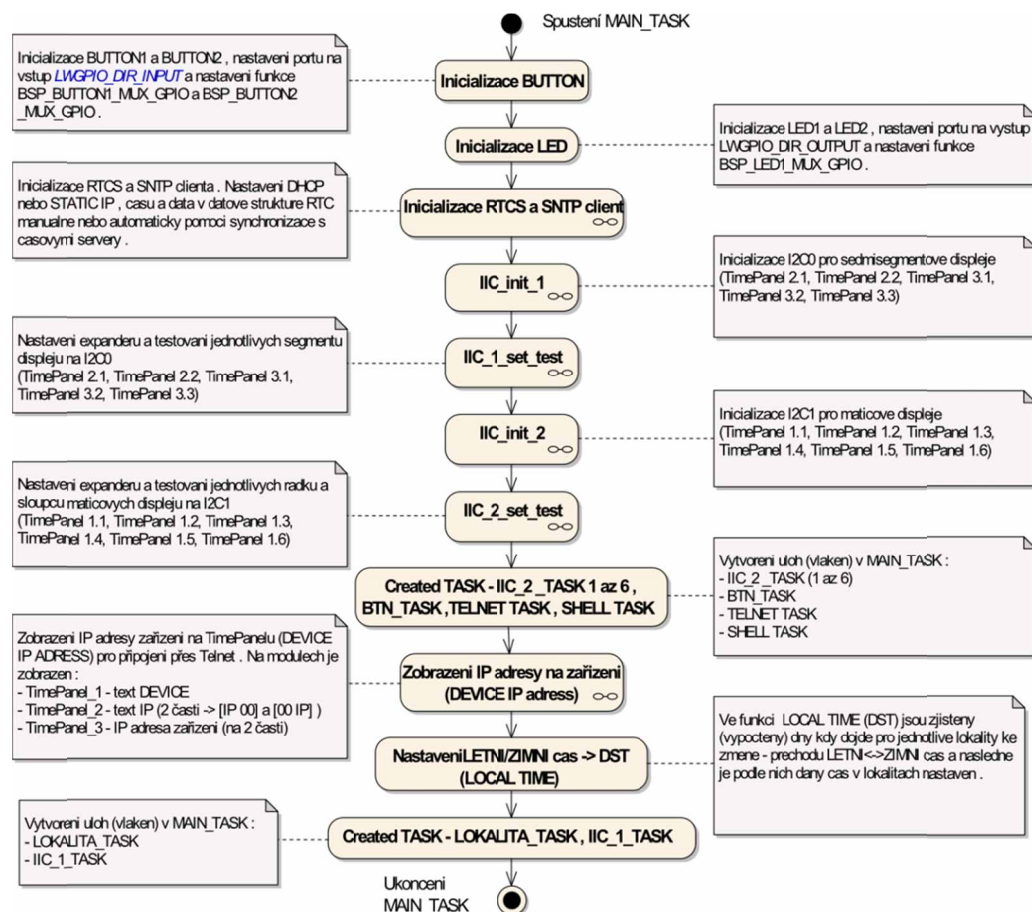
Aplikace pro TimePanel se skládá z třinácti zdrojových a hlavičkových souborů, které lze rozdělit do několika podskupin podle jejich specifikace: soubory I2C a expandérů, soubory RTCS a SNTP a soubory pro LED a tlačítka. Přehled těchto souborů je uveden v tabulce (Tabulka 5).

Název souboru	Obsah souboru
<i>buttons_irq.c</i>	Funkce pro vyčítání a práci se vstupy - tlačítka
<i>buttons_irq.h</i>	Prototypy funkcí a makra pro vstupy - tlačítka
<i>IIC_expander.c</i>	Funkce pro práci s expandérem na sběrnici I2C
<i>IIC_expander.h</i>	Makra a prototypy funkcí pro obecné I2C sběrnice
<i>leds.h</i>	Prototypy funkcí a makra pro výstupy s LED diodami
<i>leds.c</i>	Funkce pro práci a zápis na výstupní porty s LED diodami
<i>main.c</i>	Zdrojový kód jednotlivých vláken
<i>main.h</i>	Prototypy vláken a proměnných
<i>rtcs_init.c</i>	Funkce pro inicializaci RTCS
<i>rtcs_sntp.h</i>	Makra a prototypy funkcí pro RTCS a SNTP
<i>rtcs_sntp.c</i>	Funkce pro nastavení RTCS a inicializaci a spuštění SNTP
<i>lw_telnet.c</i>	Funkce pro vytvoření Telnet Serveru s příkazovým řádkem
<i>lw_telnet.h</i>	Makra a prototyp funkcí pro Telnet

Tabulka 5 Zdrojové a hlavičkové soubory aplikace TimePanel

4.4.3.1 Inicializace ‚Main_TASK‘

Úkolem úlohy ‚Main_TASK‘ je inicializace a testování periférií a modulu používaných v TimePanelu. Tato úloha je v aplikaci jako jediná vytvořená s parametrem *MQX_AUTO_START_TASK* a je tedy spuštěná automaticky po spuštění jádra MQX. Úloha slouží k nastavení, inicializaci a testování potřebných periférií (*RTCS, I2C, SNTP, GPIO...*) a vytvoření a spuštění definovaných úloh. Jednotlivé kroky úlohy ‚Main_TASK‘ jsou detailně popsány v podkapitolách a znázorněny v diagramu na obrázku (Obr. 50) a příloha (Příloha VII).



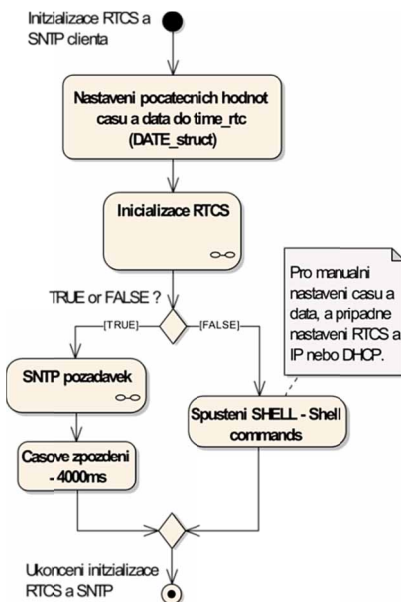
Obr. 50 Jednotlivé kroky úlohy ‚Main_TASK‘

Při spuštění úlohy ‚Main_TASK‘ nejprve dojde k inicializaci a nastavení GPIO pinu. Dva piny GPIO jsou nastaveny jako vstupní piny pro tlačítka ‚Button 1 (extern Button)‘ a ‚Button 2‘. Další dva piny GPIO jsou nastaveny jako výstupní piny pro LED diody (LED1 a LED2), které slouží k indikaci přepnutí lokality a indikaci běžícího SHELL. Další kroky této úlohy jsou již detailně popsány v podkapitolách.

Vykonání úlohy ‚Main_TASK‘ bude trvat zhruba 20 sekund (Příloha VII). Nejdéle se vykonávají funkce *IIC_1_set_test* a *IIC_2_set_test*, při kterých se postupně testují jednotlivé segmenty a tečky maticových a sedmi-segmentových displejů. U funkce *Initialization_sntp_rtcs* je doba vykonávání závislá na tom, zda uživatel vybere připojení pomocí statické nebo dynamické IP adresy. U statické IP adresy trvá vykonání funkce zhruba 500ms a v případě dynamické IP adresy je doba vykonání okolo 10 sekund.

4.4.3.1.1 Inicializace RTCS a SNTP

Úkolem úlohy „Main Task“ je inicializace RTCS a SNTP klienta a s tím spojená synchronizace a nastavení přesného času u TimePanelu. Tato inicializace je provedena pomocí funkce *Initialization_sntp_rtcs*, která se nachází ve zdrojovém souboru „rtcs_sntp.c“. Vývojový diagram jednotlivých kroků této funkce je na obrázku (Obr. 51).



Obr. 51 Inicializace RTCS a SNTP klienta

Při spuštění této funkce je nastaven pevně daný počáteční čas viz obrázek (Obr. 52) a synchronizován s MQX. Nastavení počátečního času je důležité pro ověření, zda došlo ke správné synchronizaci času TimePanelu s Time Serverem a zobrazení správného času na zařízení. Pokud se při synchronizaci nebo nastavení času vyskytne nějaká chyba, na TimePanelu se zobrazí počáteční čas, což je pro uživatele informace o tom že se vyskytla chyba.

```

void initialization_sntp_rtcs(void)
{
    printf ("\MQX synchronized to RTC time during bsp_init\n");
    /* inicializace casu a datumu v MQX */
    time_rtc.YEAR = 2012;
    time_rtc.MONTH = 12;
    time_rtc.DAY = 24;
    time_rtc.HOUR = 12;
    time_rtc.MINUTE = 36;
    time_rtc.SECOND = 0;
    time_rtc.MILLISEC = 0;

    _time_from_date (&time_rtc, &time_mqx);
    _time_set (&time_mqx);
}
  
```

Obr. 52 Nastavení počátečního stavu

U funkce *Initialization_sntp_rtcs()* je inicializace RTCS a SNTP rozdělena do dvou funkcí. Pro inicializaci a nastavení RTCS je použita funkce *rtcs_init()* viz obrázek (Obr. 53). Při spuštění této funkce dojde k nastavení proměnné *RTCS_DHCP*, pomocí které lze nastavit zda, zařízení bude pracovat se statickou IP adresou (*RTCS_DHCP 0*) nebo dynamickou (*RTCS_DHCP 1*). Dalším krokem je vytvoření RTCS, jehož návratová hodnota musí být

TRUE. V případě, že je návratová hodnota FALSE, to znamená, že při vytváření RTCS došlo k chybě (nejčastěji chyba s alokací paměti ,error 1310') a proto se funkce *rtcs_init()* okamžitě ukončí a spustí se příkazový řádek SHELL.

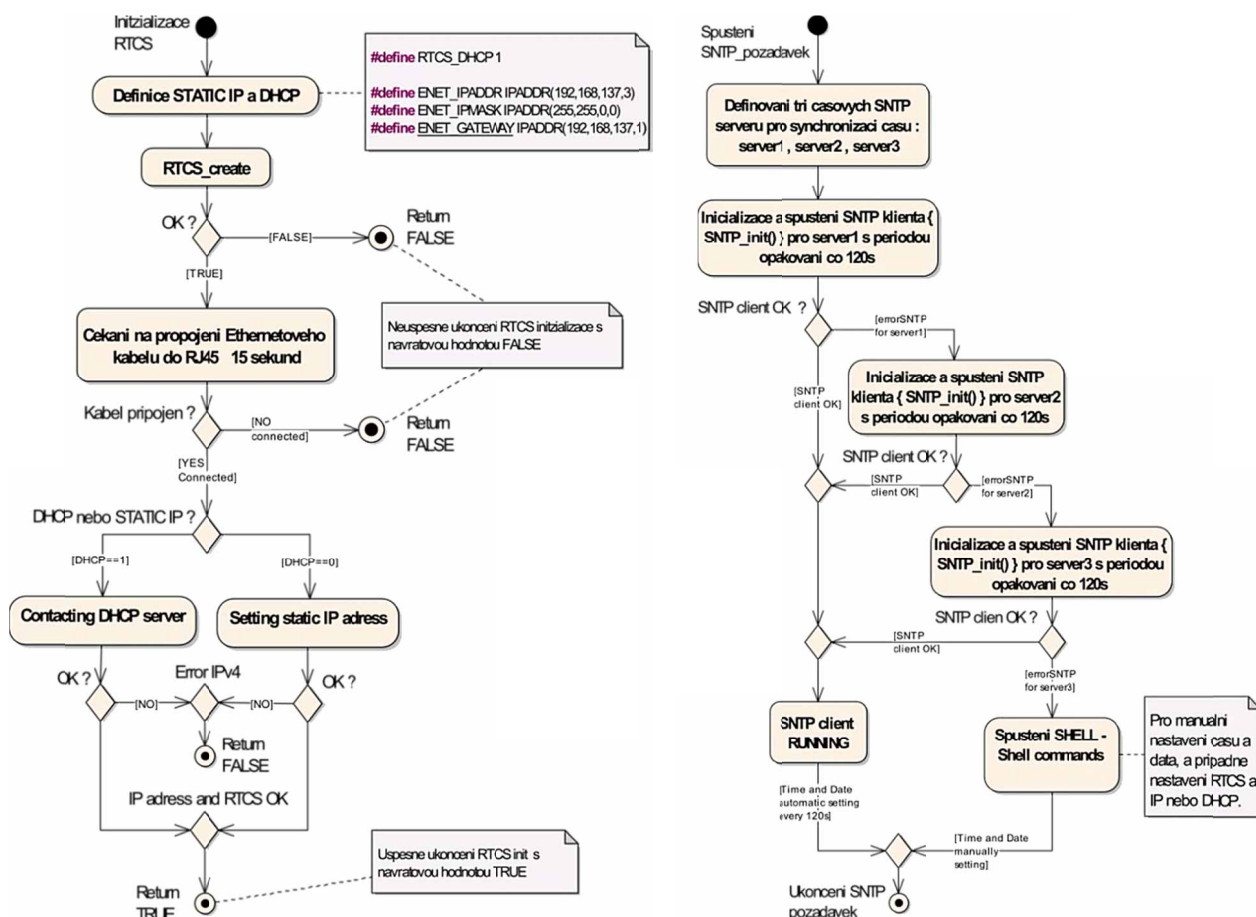
Příkaz ,RTCS_create()‘

Tato funkce alokuje zdroje, které potřebuje RTC a vytváří TCP / IP task. V případě, že chceme pracovat se sítí a internetem, je ji nutno vytvořit.

Návratové hodnoty:

- *RTCS_OK* (success)
- *Error code* (failure)

Jestliže je návratová hodnota TRUE , RTCS bylo vytvořeno a následuje 15 sekundové čekání na připojení síťového kabelu do zařízení (popřípadě sítě). Pokud není kabel připojen k zařízení nebo síti do 15 sekund, dojde opět k ukončení *rtcs_init()* a spuštění příkazového řádku SHELL. V případě, že kabel je připojen k zařízení, dojde k připojení zařízení do sítě. Pokud komunikace mezi zařízením a serverem neprobíhá v pořádku, příkazový řádek SHELL je opět spuštěn.



Obr. 53 Inicializace RTCS a synchronizace času u TimePanelu pomocí SNTP

Následně je spuštěna funkce ***SNTP_pozadavek***, která slouží k inicializaci a vytvoření vlákna *SNTP_client* pro synchronizaci času mezi zařízením a Time Serverem viz obrázek (Obr. 53). Při spuštění této funkce jsou definovány tři IP adresy Time Serveru, z kterých je získáván přesný čas viz tabulka (Tabulka 3). Dále je pomocí funkce ***SNTP_init*** vytvořeno vlákno *SNTP_client* s prioritou devět, které aktualizuje místní čas zařízení s Time Serveru *Server_1* v pravidelných intervalech. V případě, že se aktualizace času a připojení k Time Serveru *Server_1* nezdaří, následuje opětovné použití funkce ***SNTP_init*** ale již s *Server_2*. Kdyby se nepovedlo připojit k *Server_2* ani k *Server_3*, dojde ke spuštění příkazového řádku SHELL pro manuální nastavení času a data. Jestliže se připojení k vybraným Time Serverům podaří, vlákno *SNTP_client* aktualizuje místní čas zařízení s Time Servery v pravidelných intervalech.

Funkce *SNTP_init()*

Tento příkaz slouží k inicializaci SNTP klienta. Funkce spustí klienta SNTP, který se stane novým vláknem a provede první aktualizaci místního času, pak bude čekat na dosažený čas v sekundách, jak je uvedeno v parametrech. Jakmile tato doba uplyne, SNTP klient opakuje stejný cyklus. Místní čas je nastaven na UTC 0(Coordinated Universal Time).

Klient SNTP pracuje v unicast a anycast režimu. RTC poskytuje klienta SNTP, který je založený na RFC 2030 (Simple Network Time Protocol). Klient SNTP nabízí dvě různá rozhraní. Jedno slouží jako volání funkce, která nastaví aktuální čas, a další rozhraní, je úkol, který aktualizuje místní čas v pravidelných intervalech.

Funkce : ***SNTP_init*** (*"Name"*, *Priority*, *Stacksize*, *Destination*, *Time server poll*)

Parametry: *Name [in]* — Jméno tasku SNTP klienta.

- *Priority [in]* — Priorita tasku SNTP klienta.
- *Stacksize [in]* — Zásobník pro klienta.
- *Destination [in]* — IP adresa, kde zasílá SNTP požadavek o synchronizaci času
- *Time server poll [in]* — Čas mezi obnovou spojení, (1 až 4294967 sekundy).

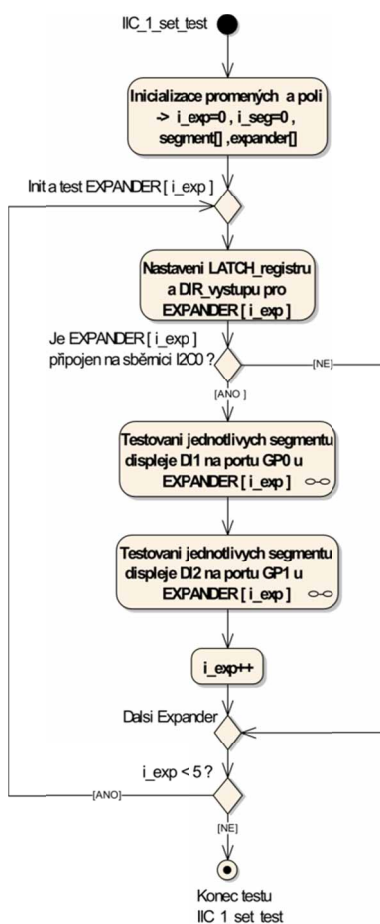
Příklad :

```
void SNTP_pozadavek(void)
{
    uint_32 errorSNTP;
    _ip_address server1;
    server1=0xC38CFED9;
    errorSNTP = SNTP_init("SNTP client", 7, 1000, server1, 120);
    if (errorSNTP) printf("SNTP client error : Server1\n");
}
```

4.4.3.1.2 Inicializace a testování modulu TimePanelu na sběrnici I2C0

Táto část úlohy *Main Task* slouží k inicializaci a nastavení sběrnice I2C0 a k otestování modulu TimePanel (2.1 až 3.3) připojených na sběrnici I2C0. Pro počáteční inicializaci a nastavení sběrnice I2C0 je použita funkce ***IIC_init_I()*** viz příloha (Příloha VII). V této funkci je otevřen ovladač sběrnice I2C0 s ukazatelem *fd_IIC_1* pomocí funkce ***fopen***, nastaví se přenosová rychlost I2C0 na 100kHz a řídicí jednotka (*PXN20*) je nastavena jako *Master* na sběrnici.

Poté se spustí funkce **IIC_1_set_test(500)**, ve které se nastaví expandéry na modulech TimePanel (2.1 až 3.3) a postupně se otestují jednotlivé segmenty displejů těchto modulu viz obrázek (Obr. 54). Po spuštění této funkce se vytvoří statické pole s pěti pevně danými adresami expandérů použitými na sběrnici I2C0 viz obrázek (Obr. 48). V dalším kroku je vybrána první adresa expandérů z pole **EXPANDER[i_exp]** a port 0 a 1 vybraného expandéru se nastaví do výstupního režimu pomocí registru **IODIR0** a **IODIR1** (*IO direction*) zapsáním log. 0 do těchto registru. Poté jsou nastaveny jeho výstupní porty (0 a 1) do log'0 zapsáním dat do výstupních registrů **OLATH0** a **OLATH1** (*output latch register*). Dále se ověří, zda je expandér s adresou **EXPANDER[i_exp]** připojen na sběrnici I2C0 a komunikuje s řídicí jednotkou. Následně může dojít k testování jednotlivých segmentů displejů na modulu s expandérem **EXPANDER[i_exp]**. V případě, že expandér **EXPANDER[i_exp]** není připojen na sběrnici I2C0 nebo nekomunikuje s řídicí jednotkou, začne se testovat další expandér a v příkazovém řádku je vypsána informace o problému s daným expandérem.

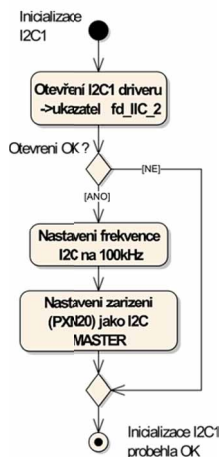


Obr. 54 Nastavení expandérů a testování segmentů displejů na modulech TimePanelu

Test jednotlivých segmentů je proveden nejdříve na sedmi segmentovém displeji DI1 a poté na DI2. Testování probíhá tak, že se na displeji rozsvítí vždy jen jeden segment na 50ms, po uplynutí tohoto časového intervalu daný segment zhasne a rozsvítí se další segment displeje. Tento test je proveden pro všechny segmenty na displeji (*segment ,A‘ až ,G‘ + ,DP‘*). Časový interval svícení jednoho segmentu je nastaven na 50ms ale lze jej snadno přenastavit zápisem nové hodnoty do vstupní proměnné funkce **IIC_1_set_test(časovy_interval)**.

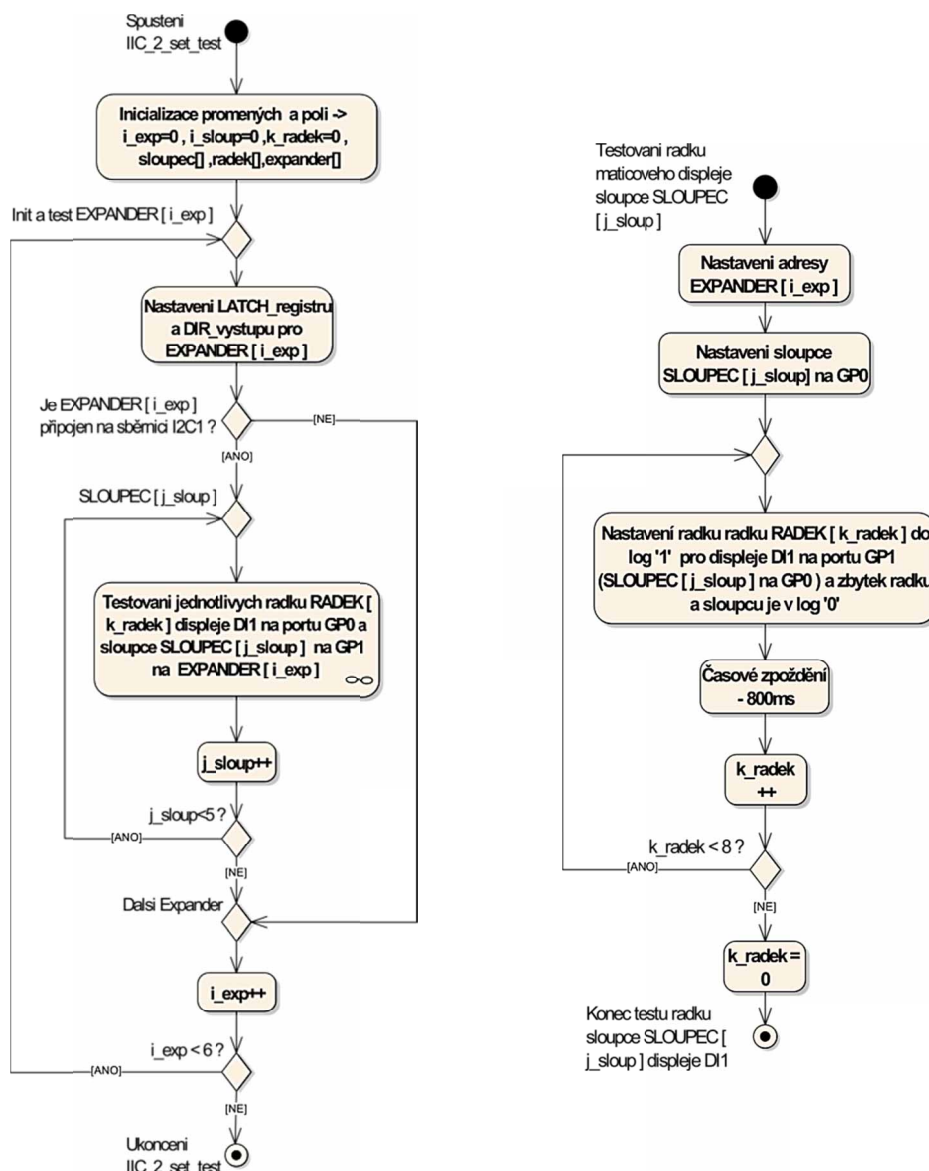
4.4.3.1.3 Inicializace a testování modulu TimePanelu na sběrnici I2C1

Dalším krokem úlohy „Main Task“ je inicializace a nastavení sběrnice I2C1 a otestování modulu TimePanel (1.1 až 1.6) připojených na sběrnici I2C1. Pro počáteční inicializaci a nastavení sběrnice I2C1 je použita funkce **IIC_init_2()** viz obrázek (Obr. 55). V této funkci je otevřen ovladač sběrnice I2C1 s ukazatelem *fd_IIC_2*, nastavená se přenosová rychlost I2C1 na 400kHz a řídicí jednotka (PXN20) je nastavena jako „Master“ na sběrnici.



Obr. 55 Inicializace a nastavení sběrnice I2C1

Po dokončení inicializace a nastavení sběrnice I2C1 se spustí **funkce IIC_2_set_test(10)**, ve které se nastaví expandéry na modulech TimePanel (1.1 až 1.6) a postupně se otestují jednotlivé body (*tečky*) maticových displejů těchto modulu viz obrázek (Obr. 56). Opět je zde použito statické pole s pevně danými adresami expandérů použitých na sběrnici I2C1 viz obrázek (Obr. 48). V dalším kroku je vybrána první adresa expandérů z pole, *EXPANDER[i_exp]* a porty 0 a 1 zvoleného expandéru se nastaví do výstupního režimu pomocí registru **IODIR0** a **IODIR1** zapsáním log. 0 do těchto registrů. Poté jsou nastaveny jeho výstupní porty (0 a 1) do log‘0 zapsáním dat do výstupních registrů **OLATH0** a **OLATH1**. Dále se ověří, zda je expandér s adresou *EXPANDER[i_exp]* připojen na sběrnici I2C1 a komunikuje s řídicí jednotkou. Následně může dojít k testování jednotlivých bodů maticového displeje na modulu s expandérem *EXPANDER[i_exp]* viz obrázek (Obr. 56). Pokud expandér *EXPANDER[i_exp]* není připojen na sběrnici I2C1 nebo nekomunikuje s řídicí jednotkou, začne testování dalšího expandéru a v příkazovém řádku je vypsána informace o problému s daným expandérem.



Obr. 56 Nastavení expandérů a testování bodů maticových displejů na modulech TimePanelu

Testování jednotlivých bodů maticového displeje DI1 na modulu s expandérem *EXPANDER[i_exp]* probíhá tak, že se na displeji rozsvítí vždy jen jeden bod *RADEK[k_radek]* vybraného sloupce *SLOUPEC[i_sloup]* na 10ms. Po uplynutí tohoto časového intervalu daný bod zhasne a rozsvítí se další řádek maticového displeje viz obrázek (Obr. 56). Tento test je postupně proveden pro každý ze sedmi řádků u všech pěti sloupců maticového displeje. Poté co je proveden test modulu s displejí na sběrnici I2C0 a I2C1 jsou vytvořeny ulohy „*BTN_TASK* a *IIC_2_TASK*“ potřebné pro vyčtení hodnoty z tlačítek a zobrazení textu na maticových displejích.

4.4.3.1.4 Zobrazení IP adresy zařízení

Následně je pomocí funkce **DEVICE_IP_ADDRESS** zjištěna IP adresa přiřazena zařízení, která je rozdělena na jednotlivé byte (Obr. 57) a zobrazena ve dvou krocích na modulech TimePanel_3. V prvním kroku je zobrazena na TimePanel_3 jedna část IP adresy a po stisku tlačítka „BTN_1“ druhá část IP adresy. Na modulech TimePanel_1 je zobrazen text „**DEVICE**“ a na modulech TimePanel_2 je zobrazen text „IP __“ v prvním kroku a v druhém kroku „__ IP“ označující aktuálně zobrazenou část IP adresy na TimePanel_3 viz obrázek (Obr. 58).

```
printf("\nIP Address 4d      : %d.%d.%d.%d\n", IPBYTES(ip_data.ip));  
//rozdeleni IP adresy 192.168.137.23 na 4 casti , po bytech  
adresa[0]=(((ip_data.ip)>>24)&0xFF); //192  
adresa[1]=(((ip_data.ip)>>16)&0xFF); //168  
adresa[2]=(((ip_data.ip)>>8)&0xFF); //137  
adresa[3]=(((ip_data.ip)&0xFF); //23
```

Obr. 57 Rozdělení IP adresy na 4 části

Pokud je zobrazena druhá část IP adresy a je stisknuté tlačítko „BTN_1“, dojde k ukončení tohoto zobrazení, nastavení letního/zimního času, vytvoření zbylých úloh a následnému zobrazení času, data a názvu lokality na TimePanelu.



Obr. 58 IP adresa zařízení zobrazena na TimePanelu- část 1 a 2

Funkce „i2c_write()“

Tento funkce „i2c_write()“ se používá k zápisu dat na expandéry po sběrnici I2C. Struktura a posloupnost zapisovaných dat je přizpůsobena expandéru MCP23016. Jedná se o funkci vytvořenou přímo pro aplikaci TimePanel .

Funkce : **i2c_write (fd ,IIC_device_adress ,registers_adress ,data_1 ,data_2)**

Parametry:

- *fd[IN]*- ukazatel na I2C kanál
- *IIC_device_adress[IN]* – adresa expandéru, na který se bude zapisovat– vstupní adresa 0x20 až 0x27
- *registers_adress[IN]* - Command byte - adresa registru v expandéru, do kterého se budou zapisovat data – vstupní adresa od 0x00 do 0x11
- *data_1[IN]* - data zapisována na GP0/GP1 port expandéru– vstupní data od 0x00 do 0xFF
- *data_2[IN]* - data zapisována na GP0/GP1 port expandéru– vstupní data od 0x00 do 0xFF

Příklad:

```
i2c_write(fd_IIC_1,0x24,0x02,0xFA,0x4B); //CMD=0x02 -> GP0 a pak GP1
```


4.4.3.2 SHELL_Task a TELNET_Task

Úloha „*SHELL Task*“ je využívána v aplikaci TimePanel pro komunikaci mezi uživatelem a TimePanelem pomocí příkazového řádku SHELL systémů MQX. Tato úloha obstarává spuštění příkazového řádku na rozhraní definovaném pomocí makra *BSP_DEFAULT_IO_CHANNEL* v konfiguračním souboru systémů „*user_config.h*“. Z důvodu snadného zobrazení a komunikace s aplikací je zvoleno rozhraní TTYB, které odpovídá sériovému rozhraní *mini USB* vyvedenému na řídicí jednotce TWR-PXN20.

Parametry spojení jsou následující:

- rychlost komunikace: **115200 baud**
- parita: **žádná**
- datové bity: **8**
- handshaking: **Xon/Xoff**

Samotný příkazový řádek SHELL je spuštěn pomocí funkce *Shell()*, které je jako parametr předána konstantní struktura obsahující požadované příkazy a jim odpovídající volané funkce. V aplikaci TimePanel tvoří hlavní skupinu používaných příkazů příkazy pro RTCS a vlastní příkazy. Přehled používaných příkazů z SHELL v aplikaci je uveden v tabulce (Tabulka 6).

<i>Příkaz</i>	<i>Volaná funkce</i>	<i>Popis</i>
"dir"	Shell_dir	Příkaz pro výpis obsahu adresáře.
"exit"	Shell_exit	Příkaz pro ukončení shellu.
"help"	Shell_help	Příkaz pro výpis nápovědy.
"?"	Shell_command_list	Příkaz pro výpis všech ostatních příkazů včetně krátké nápovědy.
"time_set"	time_shell_set	Vlastní příkaz pro manuální nastavení času v MQX.
"date_set"	date_shell_set	Vlastní příkaz pro manuální nastavení data v MQX.
"ipconfig"	Shell_ipconfig	Příkaz pro přenastavení TCP/IP (DHCP,STATIC IP adres ...).
"ping"	Shell_ping	Příkaz pro prověření funkčnosti spojení v síti.
"time_offset set"	summer_winter_ time_shell_set	Vlastní příkaz pro nastavení letního nebo zimního času na TimePanelu.
"switching_mode_set"	switching_mode_ shell_set	Vlastní příkaz pro výběr automatického režimu přepínání lokalit AUTO 1 nebo AUTO 2.
"push_button"	mode_ shell_set	Vlastní příkaz pro SW stisk tlačítka BTN1 – „ <i>dlouhý</i> “ nebo „ <i>krátký</i> “ stisk.

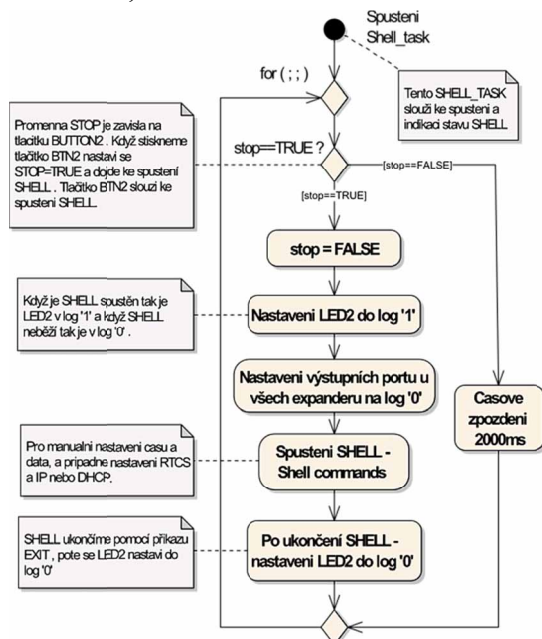
Tabulka 6 Nejpoužívanější příkazy SHELL

Ke spuštění příkazového řádku SHELL dojde v těchto případech:

- Není-li připojen síťový kabel (zařízení - síť)
- Problém s internetovým připojením a komunikací (zařízení – síť)
- Po stisku tlačítka „BUTTON_2“
- Po připojení k TELNET Serveru

Funkce „<i>Shell()</i>“
Funkce „ <i>Shell()</i> “ slouží ke spuštění příkazového řádku „<i>Shell</i>“ . Této funkci je jako parametr předána konstantní struktura obsahující požadované příkazy a jim odpovídající volané funkce. Přehled používaných příkazů z SHELL v aplikaci je uveden v tabulce (Tabulka 6). Příkazy z této tabulky byly vytvořeny pro aplikaci TimePanel. Jejich detailní popis se nachází v podkapitolách.
Příklad : <i>Shell ()</i>

Úloha „*SHELL Task*“ běží v nekonečné smyčce a slouží pro spuštění příkazového řádku pomocí tlačítka „*BUTTON_2*“ viz obrázek (Obr. 60). V případě, že při běhu aplikace dojde ke stisknutí tlačítka „*BUTTON_2*“, nastaví se proměnná *STOP* na hodnotu „*TRUE*“. Důsledkem toho je rozsvícena led dioda „*LED2*“, dojde k nastavení výstupních portů všech expandérů na log „0“ a spuštění příkazového řádku *SHELL*. Po **ukončení** *SHELL* pomocí příkazu *exit* se proměnná *STOP* nastaví do „*FALSE*“ a led dioda „*LED2*“ zhasne.



Obr. 60 Úloha „*SHELL Task*“

V úloze „*TELNET_Task*“ se vytvoří Telnet Server využívající TCP/IP protokol, který uživateli poskytuje síťový virtuální terminál pro komunikaci se zařízením. K serveru a zařízení se lze připojit pomocí IP adresy zařízení na **portu 23**. Adresa zařízení (IP) je zobrazena na displejích při spuštění TimePanelu. Při komunikaci se zařízením pomocí Telnet jsou používány příkazy z tabulky (Tabulka 6) pro běžného uživatele. Výhodou tohoto přístupu je, že **není** zapotřebí před připojením k TimePanelu provést demontáž plexiskla ze zařízení a připojení k sériové lince řídicí jednotky pomocí kabelu USB. Pomocí Telnetu je možné se k zařízení připojit/odpojit v libovolnou chvíli jen musí být TimePanel připojen do sítě přes síťový kabel.

Příkaz „*date_set*“

Příkaz „*date_set*“ je použit pro manuální nastavení data v TimePanelu **pomocí příkazového řádku SHELL**. V případě, že nejsou hodnoty vstupních parametrů zapsány do příkazu v daných rozsazích, tak funkce „*date_set*“ vypíše chybu o špatných vstupních datech a zobrazí nápovědu.

Funkce : **date_set DD MM YEAR**

Parametry : *DD[in]* - den - vstupní hodnota od 1 do 31
 • *MM[in]* – měsíc - vstupní hodnota od 1 do 12
 • *YEAR[in]* - rok - vstupní hodnota od 2000 do 2100

Příklad: *date_set 16 12 2012*

Příkaz ,time_set‘

Příkaz ,time_set‘ je použit pro manuální nastavení času v TimePanelu **pomocí příkazového řádku SHELL**. V případě, že nejsou hodnoty vstupních parametrů zapsány do příkazu v daných rozsazích, tak funkce ,time_set‘ vypíše chybu o špatných vstupních datech a zobrazí nápovědu.

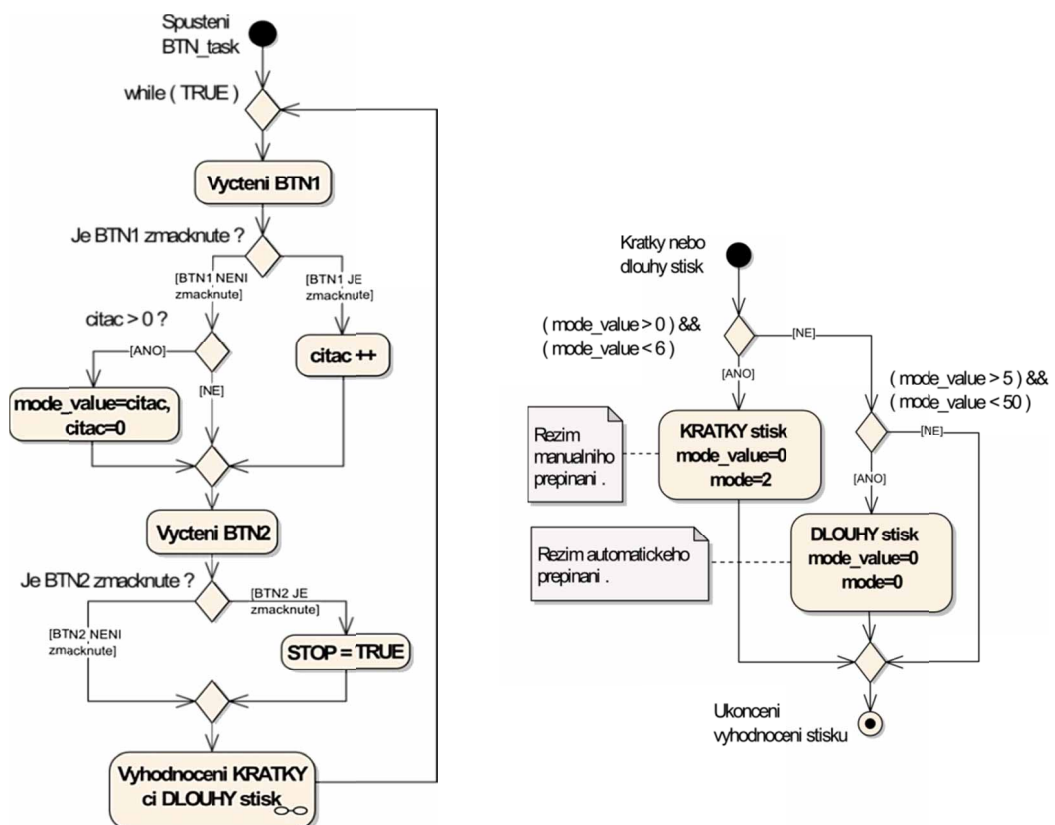
Funkce : **time_set HH MM SS**

Parametry : HH[in] - hodiny - vstupní hodnota od 0 do 23
• MM[in] – minuty - vstupní hodnota od 0 do 59
• SS[in] - sekundy - vstupní hodnota od 0 do 59

Příklad: time_set 16 12 53

4.4.3.3 BUTTON Task

Úloha ,BUTTON Task‘ slouží k zjištění stavu tlačítka ,BTN_1‘ a ,BTN_2‘ a také k určení zda se jedná o ,dlouhý‘ či ,krátký‘ stisk tlačítka viz kapitola (Kapitola 5.3.2). Tlačítko ,BTN_1‘ je použito pro přepínání režimu a lokalit u TimePanelu . Tlačítko ,BTN_2‘ slouží ke spuštění příkazového řádku SHELL. Oba tyto tlačítka jsou vyvedeny z řídicí jednotky TWR-PXN20 .



Obr. 61 Úloha ,BUTTON Task‘ a vyhodnocení stisku tlačítka - dlouhý či krátký

Pomocí periodického vyčítání hodnot (co 200ms) z tlačítek lze zjistit, v jakém stavu se nachází viz obrázek (Obr. 61). K vyčtení tlačítka ,BTN_1‘ je použita funkce *btn1_read_pin()*. Je-

li tlačítko „stisknuté“ dochází k inkrementaci čítače. Po uvolnění tlačítka je hodnota čítače předána proměnné *mode_value*, která je nutná k vyhodnocení „dlouhého“ či „krátkého“ stisku tlačítka viz obrázek (Obr. 61), a následně je čítač vynulován. Když je hodnota proměnné *mode_value* **menší než 6**, což odpovídá stisku tlačítka na dobu **kratší než 2 sekundy**, stisk je vyhodnocen jako „krátký“, proměnná ***mode* = 2** a v úloze „*LOKALITA Task*“ dojde k **přepnutí lokality v manuálním režimu**. Je-li tlačítko „*BTN_1*“ stisknuto na dobu **delší jak 2 sekundy** (hodnota *mode_value* je **větší než 6**), stisk je vyhodnocen jako „dlouhý“, nastaví se proměnná ***mode* = 0** a v úloze „*LOKALITA Task*“ se nastaví **automatický režim přepínání lokalit**.

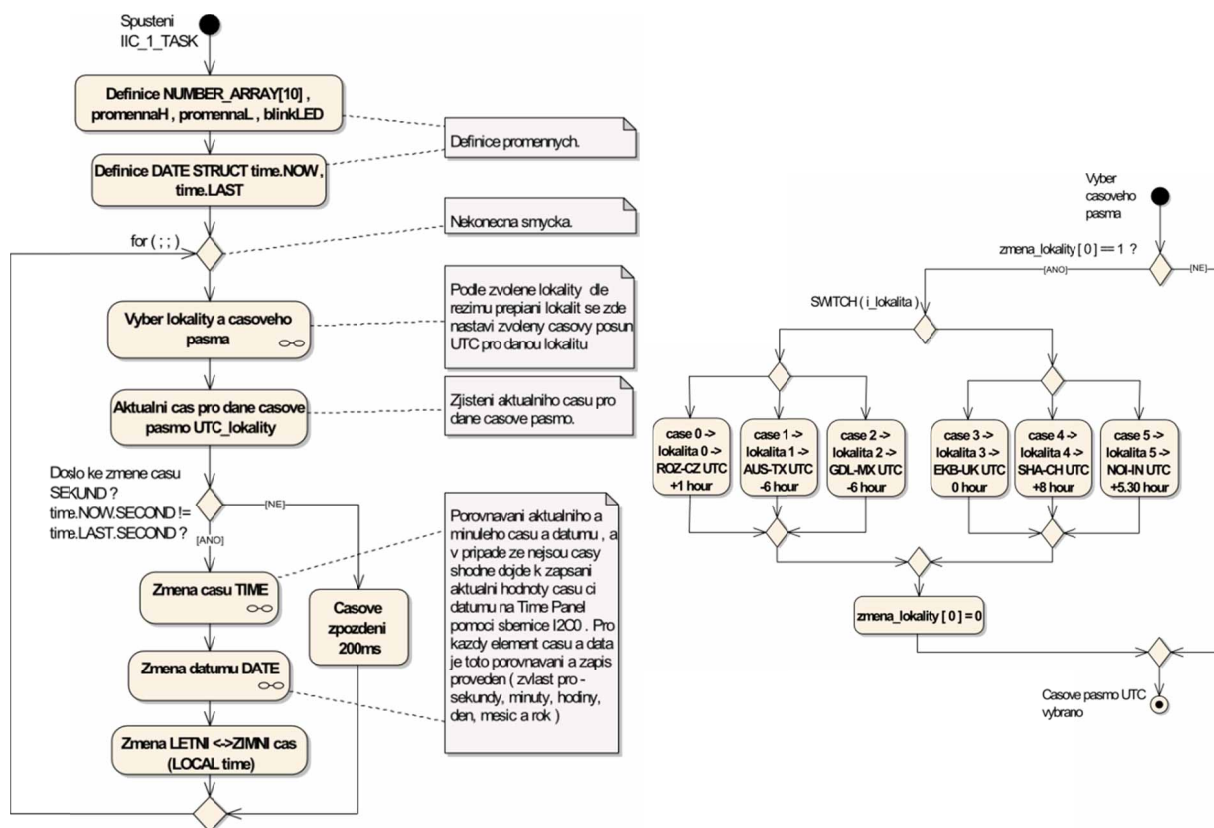
Příkaz „ <i>push_button</i> “	
Příkaz „ <i>push_button</i> “ je softwarový stisk tlačítka, obdoba mechanického stisku tlačítka <i>BTN1</i> pomocí příkazového řádku SHELL. Tímto příkazem je možné softwarově provést „dlouhý“ či „krátký“ stisk tlačítka z příkazového řádku a přepínat mezi režimy a lokalitami. V případě, že nejsou hodnoty vstupních parametru zapsány do příkazu v daných rozsazích, funkce „ <i>push_button</i> “ vypíše chybu o špatných vstupních datech a zobrazí nápovědu této funkce .	
Funkce :	<i>push_button MODE</i>
Parametry :	<i>MODE[in]</i> – režim automatického přepínání - vstupní hodnota 1 nebo 2 <ul style="list-style-type: none"> ○ 0- AUTO- automatické přepínání lokalit - dlouhý stisk ○ 1- MAN- manuální přepnutí lokality - krátký stisk
Příklad:	<i>push_button 0</i>

Pokud tlačítko „*BTN_1*“ není stisknuté a čítač je roven nule, tak úloha přechází k dalšímu kroku a to vyčtení tlačítka „*BTN_2*“ pomocí funkce ***btn2_read_pin()***. Když je tlačítko „*BTN_2*“ stisknuté, nastaví se proměnná ***stop* = TRUE** a dojde ke spuštění SHELL. V případě, že tlačítko není stisknuté, je proměnná ***stop* = FALSE** a příkazový řádek čeká na spuštění. Proměnná *stop* je využívána v úloze „*SHELL Task*“.

4.4.3.4 IIC_1 Task

Úloha „*IIC_1 Task*“ je využívána pro zobrazování času a data na TimePanelu. Hlavním úkolem této úlohy je posílání dat s aktuálním časem a datem podle zvolené lokality a časového pásma, po sběrnici I2C0 na expandéry modulu TimePanel 2.1 až 3.3. Vývojový diagram této úlohy je na obrázku (Obr. 62).

Při spuštění této úlohy se nejprve definují proměnné jako například: statické pole *NUMBER_ARRAY[10]* znaků 0 až 9, datové struktury *time.NOW* a *time.LAST* pro uložení aktuálního a minulého času atd. Následně je spuštěna smyčka, ve které se cyklicky detekuje změna času a data (*time.LAST* != *time.NOW*) a v případě, že došlo ke změně je poslán aktuální čas a datum na expandéry modulu TimePanel 2.1 až 3.3.



Obr. 62 Úloha „IIC_1_Task“ a výběr lokality a časového pásma

Ve funkci **Výběr_lokality_a_časového_pásma** se zjistí pomocí proměnné *zmena_lokalita[0]*, zda došlo k přepnutí lokality u TimePanelu viz obrázek (Obr. 62). Jestliže došlo k přepnutí lokality, pomocí proměnné *i_lokalita* se vybere nové časové pásmo aktuální lokality a v zařízení je přenastaven čas.

V dalším kroku, funkce **print_current_time(UTC_lokalita)** získá aktuální čas a datum pro dané časové pásmo a uloží jej do proměnné pro aktuální čas *time.NOW*. Poté se cyklicky porovnávají sekundy aktuálního *time.NOW.SECOND* a minulého *time.LAST.SECOND* času a v případě, že jsou sekundy (časy) rozdílné, zapíše se aktuální čas a datum na dané moduly TimePanelu. V úloze je čas a datum rozdělen na dílčí části, kdy se zvlášť sekundy, minuty, hodiny, dny, měsíce a roky aktuálního a posledního času a data navzájem porovnávají. V případě, že jsou rozdílné, aktuální dílčí hodnoty času a data se zapíší na dané moduly.

Funkce **print_current_time()**

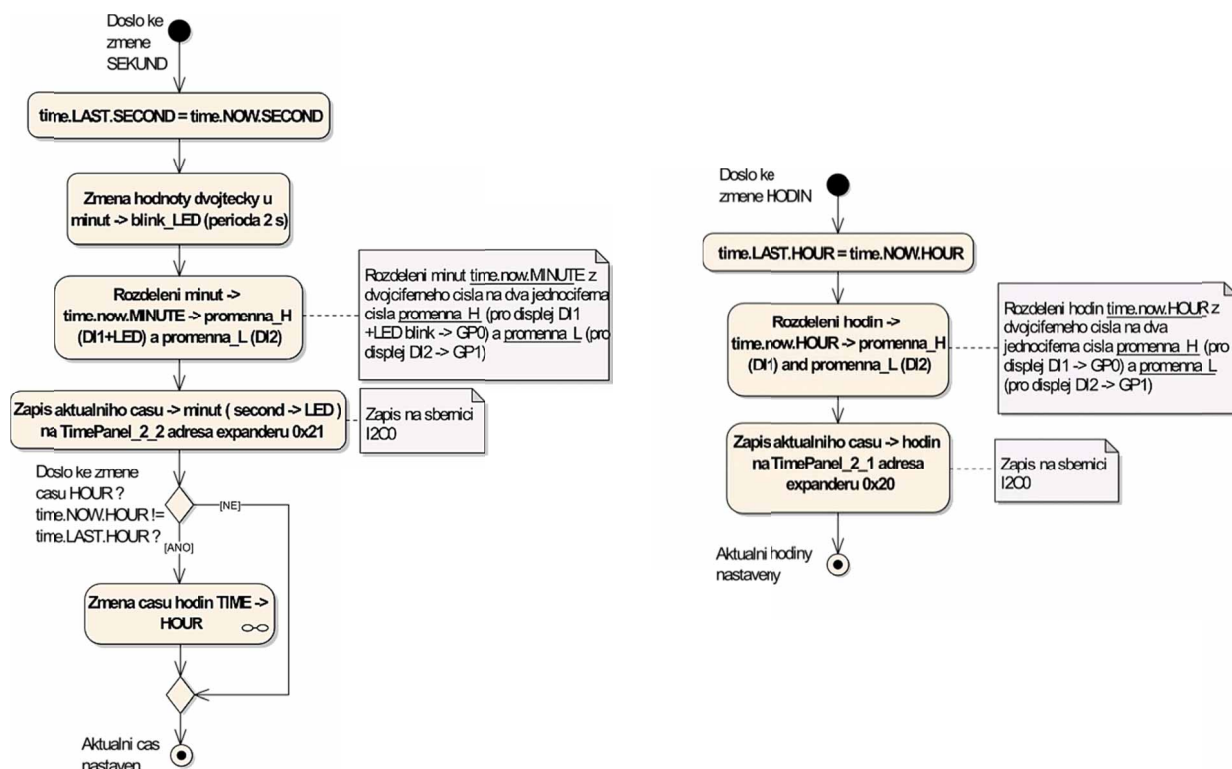
Funkce **print_current_time()** se používá pro zjištění aktuálního přesného času s daným časovým posunem UTC v MQX. Tato funkce následně uloží hodnotu zjištěného aktuálního času s časovým posunem do globální proměnné *time_rtc*, se kterou se pak v aplikaci po vyčtení času pracuje.

Funkce: **print_current_time (UTC)**

Parametry:

- *UTC [in]* – časový posun UTC v sekundách – vstupní hodnota od -43200 do +50400

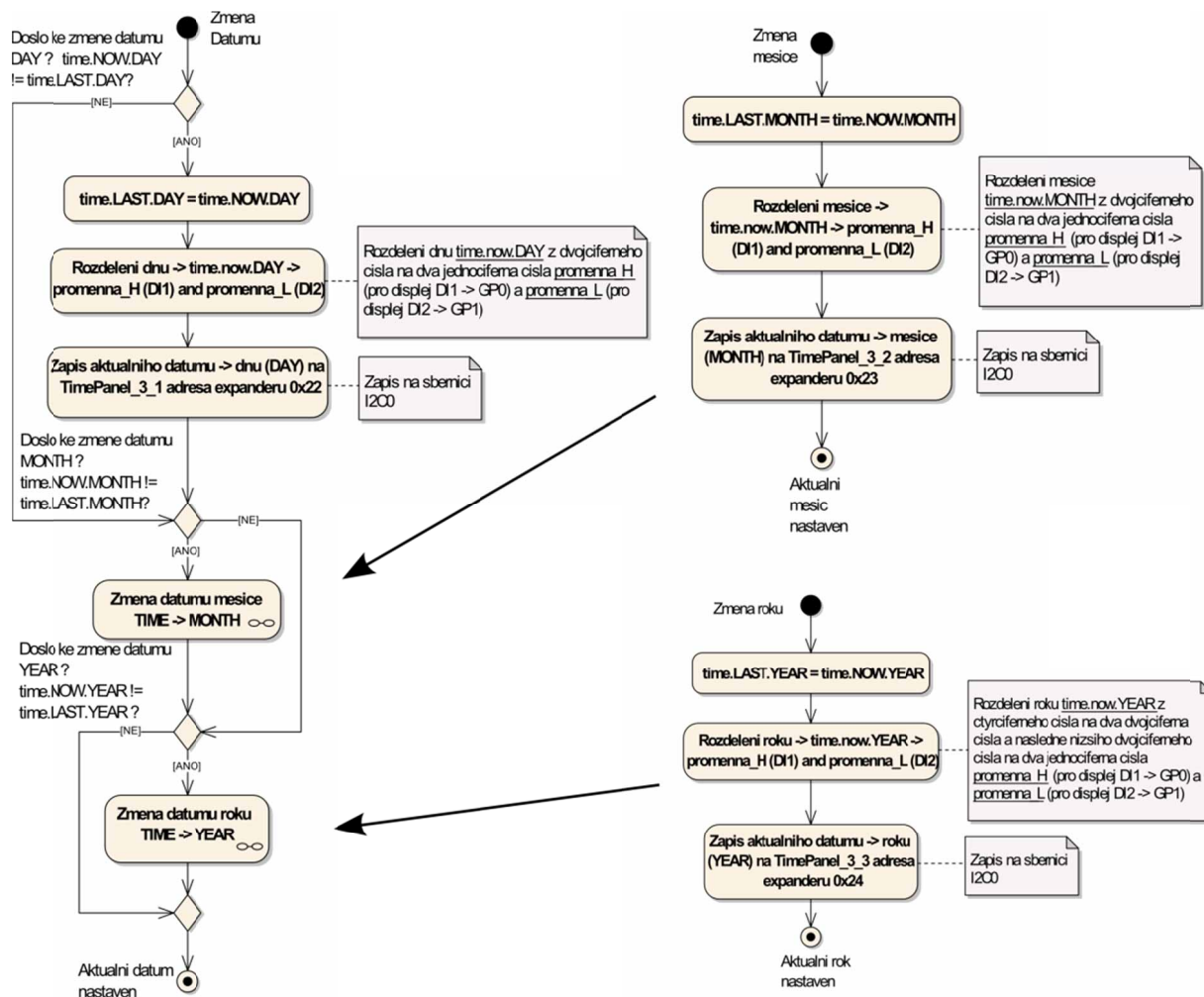
Příklad: **print_current_time (- 5400)**



Obr. 63 Funkce změna času TIME a změna času hodin – TIME - HOUR

Na modulu TimePanel 2_2, který obsahuje dva sedmi segmentové displeje pro zobrazení minut a dvě led diody pro indikaci sekund, je zapotřebí posílat data každou sekundu, i když nedojde ke změně minut. Důvodem je indikace sekundy na modulu blikáním led diod s periodou 2 sekundy. Pro zobrazení aktuálních sekund a minut na TimePanelu je použita funkce **Změna času TIME**. Tato funkce nejdříve invertuje hodnotu proměnné *blink_LED*, která se používá pro indikaci sekund a nabývá hodnot **0** a **1**. Následně se rozdělí aktuální minuty *time.NOW.minute* z dvojciferného čísla na dvě jednociferná čísla (*promenna_a* a *promenna_L*) o velikosti 1 byte, které již lze odeslat po sběrnici I2C0. Poté jsou zapsány aktuální minuty a sekundy pomocí proměnných *promenna_H* a *promenna_L* na expander modulu **TimePanel 2_2 s adresou 0x21** viz obrázek (Obr. 63). Odesílání aktuálních minut a sekund na daný modul se vykonává každou sekundu.

Jestliže došlo v předchozích krocích ke změně a odeslání sekund (minut), v následujících krocích se porovnají proměnné hodin *time.NOW.HOUR* a *time.LAST.HOUR* a jsou-li rozdílné, opět se rozdělí aktuální hodiny *time.NOW.HOUR* a zapíše se na expander modulu **TimePanel 2_1 s adresou 0x20** viz obrázek (Obr. 63). Dále se ve funkci **DTS_TIME()** porovná aktuální čas a datum s vypočtenými dny (změna **LETNÍ** < - > **ZIMNÍ** čas) a v případě, že jsou shodné, je proveden posun na letní (zimní) čas.

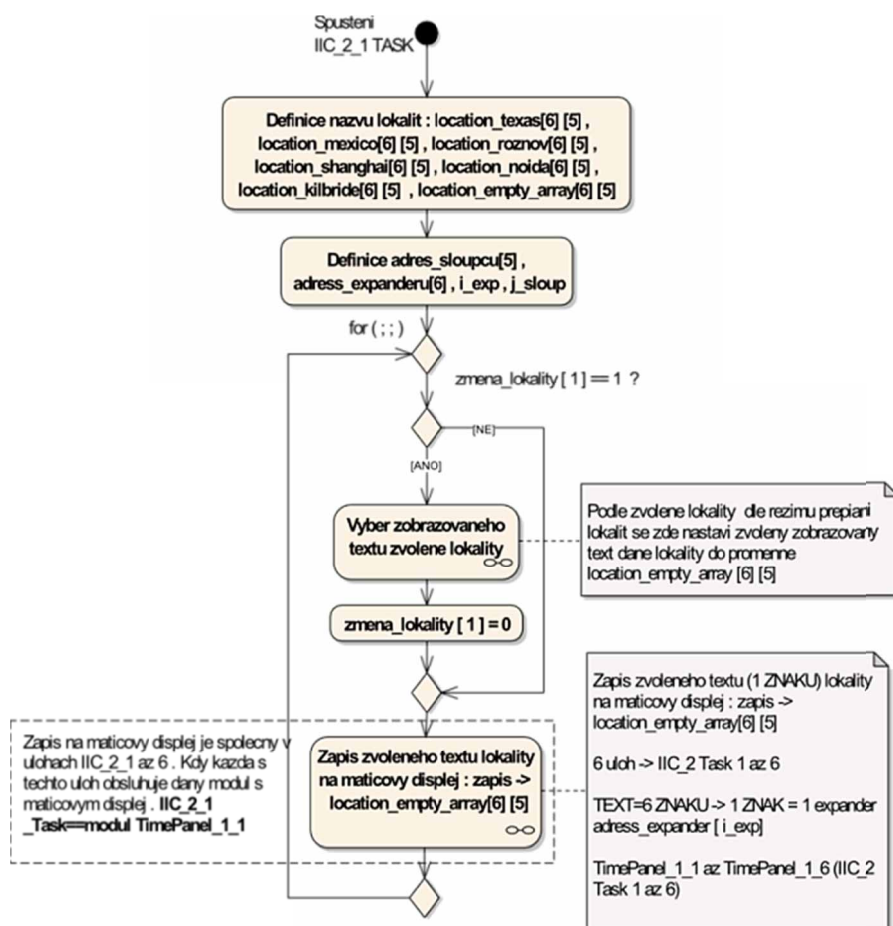


Obr. 64 Funkce pro změnu data – dnu, měsíce a roku

V dalších krocích se řeší porovnávání a zápis u dílčích částí data, kdy je postup stejný jako v předchozích krocích u času. Opět se navzájem porovnávají u dnů, měsíců i roků jejich hodnoty proměnných *time.NOW* a *time.LAST*. Před zápisem aktuální hodnoty dnů *time.NOW.day* a měsíců *time.NOW.month* se opět tyto hodnoty rozdělí z dvojčíselného čísla na dvě jednociferná (*promenna_H* a *promenna_L*) a zapíše se na expandéry modulu **TimePanel 3_1 s adresou 0x22** zobrazující den a **TimePanel 3_2 s adresou 0x23** zobrazující měsíc viz obrázek (Obr. 64). U zápisu aktuálního roku *time.NOW.year* je postup rozdělení proměnné jiný, jelikož rok je čtyřciferné číslo. Proměnná *time.NOW.year* se rozdělí na dvě dvojčíselná čísla, ze kterých se pak vybere nižší dvojčíselné představující desetiletí a tato hodnota se rozdělí na dva jednociferná čísla (*promenna_H* a *promenna_L*) a zapíše se na expandér modulu **TimePanel 3_3 s adresou 0x24** zobrazující rok. Pak jsou ve funkci *zjisteniDNUzmenyDST()* přepočteny dny, kdy dojde u jednotlivých lokalit ke změně - přechodu „LETNÍHO“ < - > „ZIMNÍHO“ času.

4.4.3.5 IIC_2_Task 1 až 6

Úlohy *IIC_2_Task 1 až 6* zobrazují název aktuální zvolené lokality na TimePanelu. Jejich úkolem je tedy posílání dat s aktuálním názvem lokality po sběrnici I2C1 na expandéry modulu TimePanel 1.1 až 1.6. Kdy každá z šesti úloh *IIC_2_Task 1 až 6* posílá data – jeden znak z aktuálního názvu na jeden zvolený modul. V úloze *IIC_2_Task 1* jsou navíc definovány názvy lokalit a funkce pro výběr a nastavení lokality. Vývojový diagram této úlohy je na obrázku (Obr. 65).



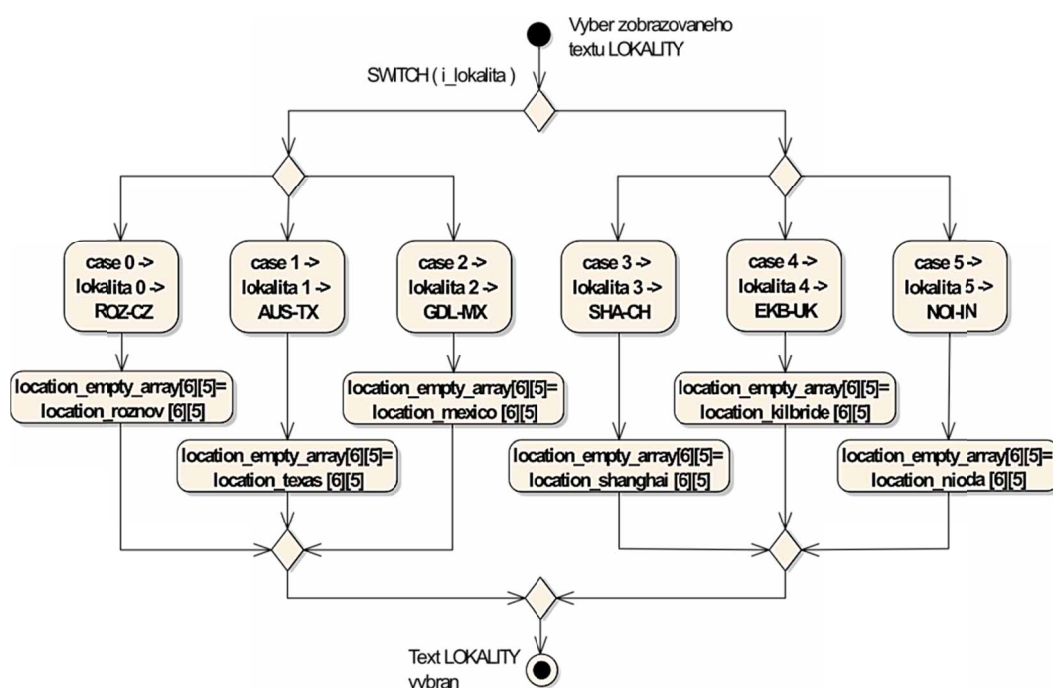
Obr. 65 Úloha *IIC_2_Task 1*

Při spuštění této úlohy se definují proměnné jako například: statické pole adres *adresy_sloupcu[5]* jednotlivých pěti sloupců maticového displeje, adresy expandérů *adresy_expanderu[6]* a další pomocné proměnné. Je také zapotřebí zapsat názvy lokalit do proměnných. Jednotlivé názvy lokalit spolu s jednou pomocnou lokalitou *empty* jsou nadefinovány pomocí sedmi dvojrozměrných statických polí s pěti sloupci a šesti řádky. Každý jeden řádek z šesti řádků definovaného 2D pole představuje jeden znak z názvu lokality a každý jeden sloupec pole představuje jeden sloupec maticových displejů viz obrázek (Obr. 66).

Modul	Řádky pole	Sloupce maticových displejů - sloupce pole					Jednotlivé znaky názvu lokality
		1	2	3	4	5	
TimePanel 1.	1	0xFE	0x90	0x98	0x94	0x62	R
TimePanel 1.	2	0x7C	0x82	0x82	0x82	0x7C	O
TimePanel 1.	3	0x86	0x8A	0x92	0xA2	0xC2	Z
TimePanel 1.	4	0x00	0x10	0x10	0x10	0x00	-
TimePanel 1.	5	0x7C	0x82	0x82	0x82	0x44	C
TimePanel 1.	6	0x86	0x8A	0x92	0xA2	0xC2	Z

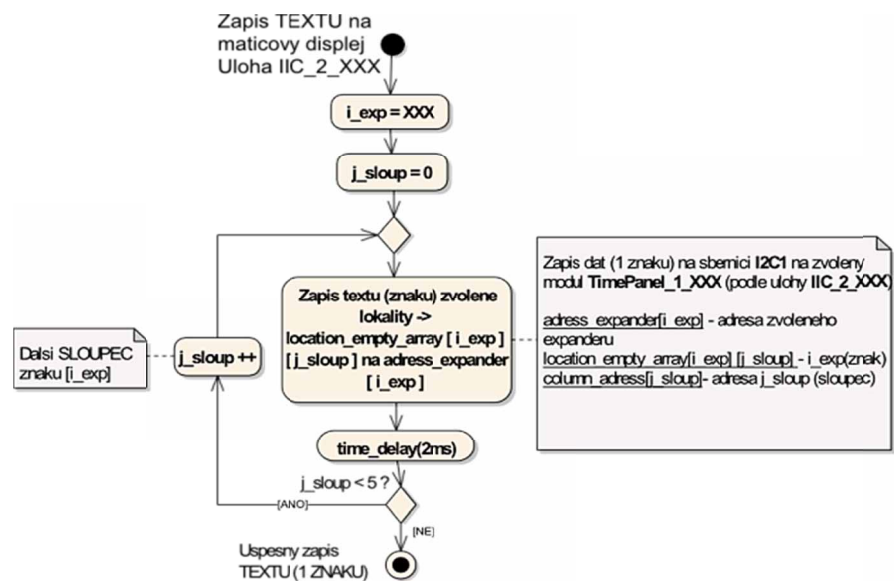
Obr. 66 Způsob definování názvů lokality do 2D pole

Ve funkci *Vyběr_zobrazovaného_názvu_lokality* se pomocí proměnné *zmena_lokality[1]* zjistí, zda došlo k přepnutí lokality u TimePanelu viz obrázek (Obr. 67) a jestliže došlo k přepnutí lokality, je vybrán nový název zobrazované lokality pomocí proměnné *i_lokalita*, který se uloží - zkopíruje do společné proměnné *location_empty_array[6][5]*.



Obr. 67 Výběr zobrazovaného názvu lokality

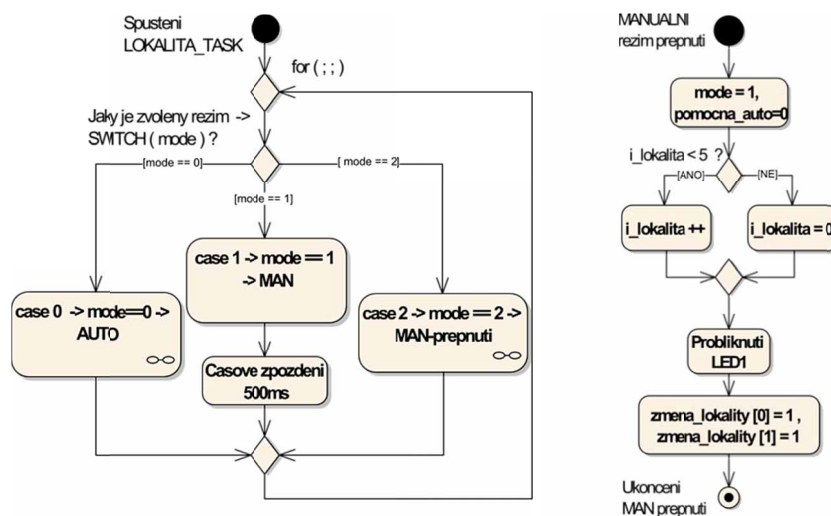
Po vybrání názvu lokality a uložení do společné proměnné *location_empty_array[6][5]* jsou data ze společné proměnné posílány úlohami *IIC_2_Task 1 až 6* po sběrnici I2C1 na moduly s maticovými displeji. Každé jedné úloze *IIC_2_Task 1 až 6* je přiřazen jeden znak ze společné proměnné, který se cyklicky zapisuje na určený modul. Zápis znaků probíhá následovně viz obrázek (Obr. 68). Ze společné proměnné je vybrán jeden řádek 2D pole (*1 ZNAK*) a postupně se cyklicky zapisuje sloupec po sloupci na určený modul s danou adresou viz obrázek (Obr. 48).



Obr. 68 Postup zápisu jednoho znaku na modul s maticovým displejem

4.4.3.6 LOKALITA Task

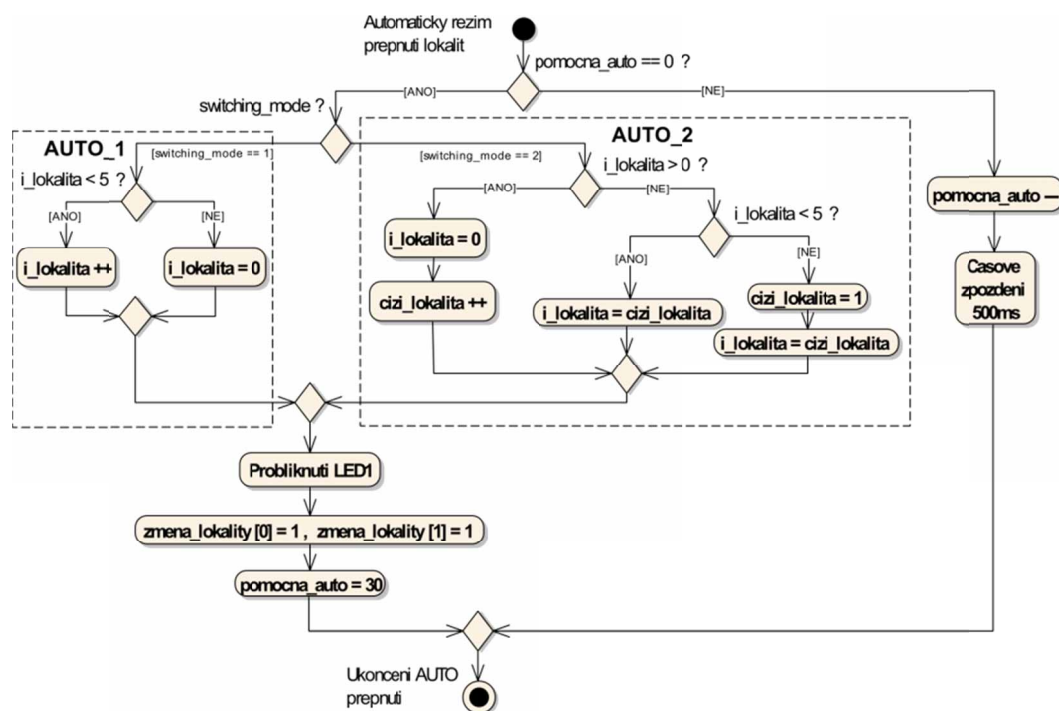
Hlavním úkolem úlohy ‚*LOKALITA Task*‘ je přepínání lokalit TimePanelu podle zvoleného režimu přepínání (*AUTO*, *MAN*). Volba režimu a přepínání lokalit je závislá na hodnotě proměnné *mode* z úlohy ‚*BUTTON Task*‘. Vývojový diagram úlohy ‚*LOKALITA Task*‘ je na obrázku (Obr. 69).



Obr. 69 Úloha ‚*LOKALITA Task*‘ a diagram přepnutí lokality v manuálním režimu

V úloze se periodicky, co 500 ms zjišťuje hodnota proměnné *mode* a pomocí podmínky *Switch(mode)* je pak vybrána větev odpovídající hodnotě této proměnné se zvoleným režimem přepínání lokalit. Je-li proměnná ‚*mode == 0*‘, je vybrán automatický režim přepínání lokalit viz obrázek (Obr. 70). V tomto režimu je vybrán pomocí proměnné *switching_mode* jeden ze dvou automatických režimů přepínání *AUTO_1* nebo *AUTO_2*. V příkazovém řádku příkazem *switching_mode_set* je možné si zvolit jeden z těchto režimů. Při inicializaci systému je nastaven režim *AUTO_2*. V automatickém režimu je pomocí proměnné ‚*pomocna_pro_auto=30*‘ nastaven časový interval (15s) zobrazení jedné lokality. Tato proměnná se dekrementuje každých

500ms, tak dlouho, dokud není ,pomocna_pro_auto == 0', pak dojde k přepnutí lokality (AUTO_1 nebo AUTO_2), probliknutí led diody ,LED1' a opětovnému nastavení časového intervalu ,pomocna_pro_auto=30'.



Obr. 70 Diagram automatického přepínání lokalit

Pokud proměnná ,mode=2', je vybrán manuální režim přepnutí lokality viz obrázek (Obr. 69), ve kterém dojde k okamžitému manuálnímu přepnutí z lokality ,A' na lokalitu ,B'. Do tohoto režimu se lze dostat jen při ,krátkém' stisku tlačítka ,BTN_1', kdy se nastaví proměnná ,mode=2'. Po manuálním přepnutí lokality a probliknutí led diody ,LED1' se nastaví proměnná ,mode=1' a je vybrán režim, ve kterém se zobrazena lokalita na panelu nemění a nedochází k přepnutí lokality. V tomto režimu zůstává tak dlouho, dokud není proveden ,krátký' nebo ,dlouhý' stisk tlačítka a tudíž přepnutí do jiného režimu.

Příkaz ,switching_mode_set'

Příkaz ,switching_mode_set' je použit pro manuální nastavení a volbu režimu automatického přepínání lokalit v TimePanelu pomocí příkazového řádku SHELL. Pomocí tohoto příkazu je možno zvolit jeden ze dvou režimů automatického přepínání – režim AUTO_1 a AUTO_2. Při inicializaci aplikace je nastaven režim AUTO_2. Detailní popis těchto režimů je v kapitole (Kapitola 5.3.2). V případě, že nejsou hodnoty vstupních parametru zapsány do příkazu v daných rozsazích, funkce ,switching_mode_set' vypíše chybu o špatných vstupních datech a zobrazí nápovědu této funkce.

Funkce: switching_mode_set MODE_AUTO

Parametry: MODE_AUTO[in] – režim automatického přepínání- vstupní hodnota 1 nebo 2
 ○ 1 - režim AUTO_1 – zobrazení 1xROZN_CZ a 1x cizí lokalita
 ○ 2 - režim AUTO_2 – všech 6 lokalit se zobrazuje postupně po sobě

Příklad: switching_mode_set 2

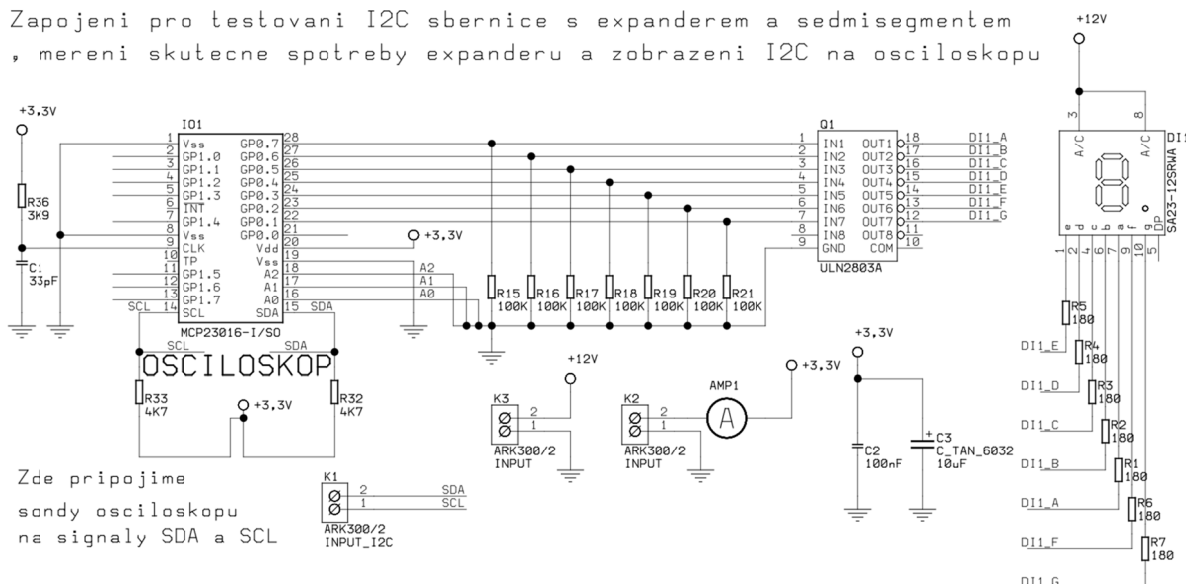
5 Testování a oživování HW a SW

V této kapitole je popsán průběh testování hardwarů a softwarů na nepájivém poli, které probíhalo před započítím realizační části řešení. Dále jsou zde popsány jednotlivé postupy při oživení a testování rozšiřující desky a modulu TimePanelu. V závěru kapitoly je shrnut průběh kompletního softwarového a hardwarového testování TimePanelu.

5.1 Předrealizační testování HW a SW

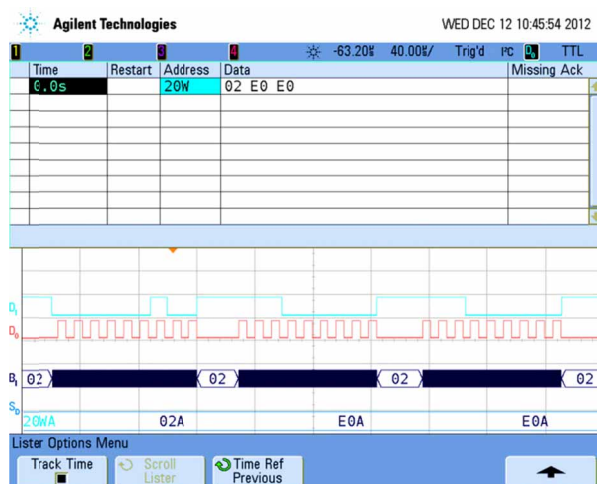
Předtím než byly jednotlivé plošné spoje LED modulů (TimePanel 1 až 3) poslány do výroby, byla ověřena funkce expandéru při napájení 3.3V na nepájivém poli. Spolu s expandérem MCP23016 bylo také otestováno tranzistorové pole UNL2803 se sedmi segmentovým displejem. K expandéru byl připojen digitální analyzátor pro sledování signálů na sběrnici I2C a ampérmetr pro měření spotřeby expandéru. Schéma testovacího zapojení expandéru je na obrázku (Obr. 71). Naměřena spotřeba expandéru byla okolo 20mA, což je mnohonásobně nižší hodnota než maximální spotřeba udávaná výrobcem.

Zapojení pro testování I2C sběrnice s expandérem a sedmisegmentem
, měření skutečné spotřeby expandéru a zobrazení I2C na osciloskopu



Obr. 71 Schéma pro měření spotřeby expandéru a testování I2C s expandérem a displejem

Při testování je k expandéru připojena sběrnice I2C0 z řídicí jednotky TWR-PXN20. V řídicí jednotce je nahrána testovací aplikace, která byla navržena pro účely testování I2C s expandérem a sedmi segmentovým displejem. Při testu se cyklicky posílají data na expandér, zobrazující na sedmi segmentovém displeji znaky 0 až 9. Na obrázku (Obr. 72) jsou pomocí digitálního analyzátoru zachyceny datové průběhy sběrnice I2C při testu. Na obrázku (Obr. 72) s daty na sběrnici I2C je vidět odeslanou adresu expandérů 0x20, data 0x02 pro nastavení latch registru (Command byte) a nakonec data 0xE0 a 0xE0 představující na displeji znak ,7'.



Obr. 72 Data na sběrnici I2C

Testovací aplikace byla dále doplněna o úlohy pro testování GPIO vstupu/výstupu, připojení k síti (*DHCP* nebo *STATIC IP*) a synchronizaci času MQX s Time Serverem (SNTP). Zařízení je možno připojit do sítě pomocí dynamické IP adresy (*RTCS_DHCP* = 1) nebo pevné IP adresy (*RTCS_DHCP* = 0 – nutno nastavit IP adresy ručně) viz obrázek (Obr. 73).

```

//notbook
#define ENET_IPADDR IPADDR(192,168,137,1)
#define ENET_IPMASK IPADDR(255,255,0,0)
#define ENET_GATEWAY IPADDR(192,168,137,3)

#define ENET_DEVICE 0
#define RTCS_DHCP 1 // DHCP == 0 -> static IP

#define PPP_DEVICE "itttyb:"

```

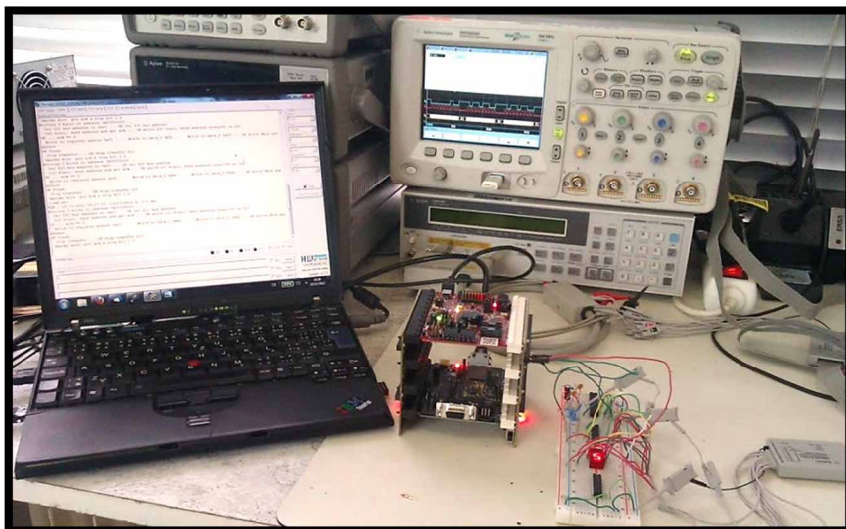
Obr. 73 Nastavení DHCP nebo STATIC_IP

Po zprovoznění připojení k síti společně se synchronizací času MQX s Time Serverem, byl následně uveden do chodu a otestován příkazový řádek SHELL s vytvořenými příkazy pro manuální nastavení času a data za běhu aplikace. Jednotlivé úlohy byly testovány při různých situacích viz obrázek (Obr. 74) s příklady.

Situace	→	Reakce systému - aplikace
Spuštění MQX bez zapojeného síťového kabelu	→	Po 15s se spustí SHELL pro nastavení času a data manuálně
Síť. kabel zapojený ale špatně IP adresy Time serveru (SNTP)	→	Spustí se SHELL pro nastavení času a data manuálně
Síť. Kabel zapojený, dobré IP adresy Time serveru (SNTP)	→	Vše OK - čas MQX synchronizovaný s Time Severem

Obr. 74 Testování SNTP, Ethernet a SHELL

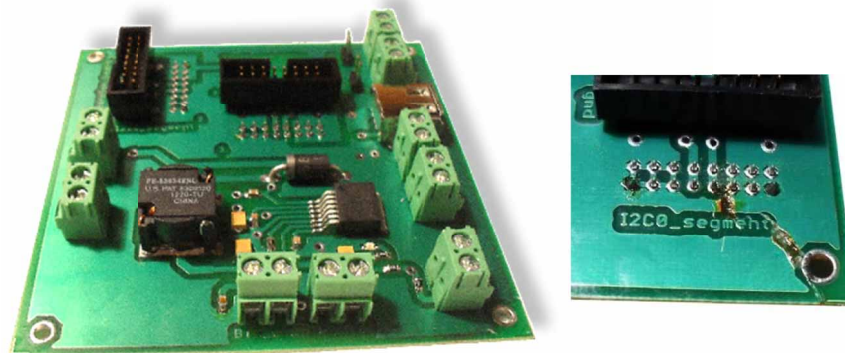
Také byla testována situace, kdy po spuštění systému proběhla úspěšná synchronizace času MQX s Time Serverem, následně byl za běhu aplikace síťový kabel odpojen a čas byl manuálně přenastaven pomocí příkazového řádku na špatný čas. Po určité době byl však síťový kabel zpět připojen do zařízení, a jelikož úloha *CLIENT_task* (*SNTP_init*) synchronizuje čas periodicky (každých 120sek), automaticky došlo k přenastavení času MQX na přesný čas s Time Serverem. Na obrázku (Obr. 75) je předrealizační testovací pracoviště.



Obr. 75 Testovací pracoviště

5.2 Rozšiřující deska „*Expansion_BOARD*“

Před osazením a oživením rozšiřující desky, bylo provedeno kontrolní měření odporu mezi jednotlivými cestami, piny a konektory na neosazené desce, a to z důvodu odhalení případné chyby na DPS. Touto kontrolou bylo zjištěno, že se na DPS nachází zkrat mezi napájením +3.3V a zemi GND. K této chybě došlo při návrhu DPS u vylévání mědi, jelikož ohraničení (rastr) vylévané mědi napájení (vrstva BOTTOM) se překrývalo s ohraničením země. To mělo za následek spojení vylévané mědi 3.3V a GND dohromady. Tato chyba byla odstraněna přerušením dvou vodičů na plošném spoji (Obr. 76).

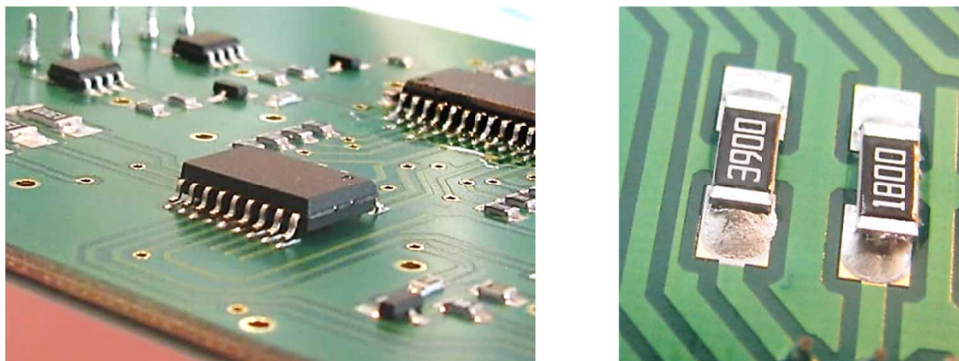


Obr. 76 Osazená rozšiřující deska a odstranění chyby

Následně byly na DPS osazeny součástky potřebné pro DC-DC měnič, který byl ihned otestován připojením k 12V zdroji a bylo změřeno výstupní napětí z měniče (3.3V). Poté byla rozšiřující deska osazena zbylými pasivními součástkami.

5.3 Moduly TimePanelu

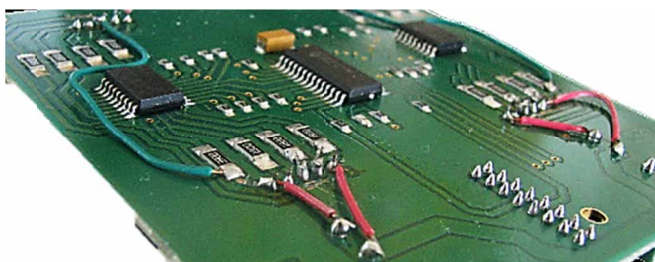
Desky plošných spojů modulů TimePanelu byly vyrobeny ve firmě *PragoBOARD s.r.o.* a osazeny firmou *ČeMeBO s.r.o.* Testování zda jsou všechny moduly funkční probíhalo na každém modulu samostatně a průběh byl následující. Testovaný modul byl připojen k řídicí jednotce pomocí sběrnice I2C a byla spuštěna testovací aplikace, ve které se nejdříve zjistilo, zda je modul správně připojen na sběrnici, následně se porty (0 a 1) expandérů nastavily pomocí registru **IODIR** do výstupního režimu a pokud vše proběhlo v pořádku, všechny segmenty (tečky) na displejích modulu byly postupně otestovány.



Obr. 77 Zjištěné závady u modulů TimePanel 1.1 až 2.2

Tímto testem prošlo zhruba 70% modulů TimePanel 1.1 až 2.2. U zbylých 30% modulů, které testem neprošly, byly zjištěny závady jako například studené spoje, neosazené součástky, nezapájené piny nebo naopak spojení více pinů dohromady u integrovaných obvodů viz obrázky (Obr. 77). Všechny moduly byly proto důkladně překontrolovány a závady opraveny.

U modulů TimePanel 3 byl zjištěn problém s osazeným displejem, místo **SA23-12SRWA** byl osazen **SA23-11SRWA**, který měl jinak rozmístěné piny. Na modulech TimePanel 3 bylo proto zapotřebí provést pár úprav, například přerušit pár vodivých cest a přemostit některé piny a cesty pomocí drátových propojek, aby modul fungoval správně i s displejem **SA23-11SRWA** viz obrázek (Obr. 78).



Obr. 78 Upravený modul TimePanel_3

5.4 Kompletní test TimePanelu

Po sestavení mechanické části (konstrukce) a zkompletizování s elektrickými částmi (moduly a DPS, zdroje a řídicí jednotka) byla společně se zařízením odlaďována aplikace (SW i HW) viz obrázek (Příloha I). Při odlaďování SW bylo například důležité vhodně nastavit časový interval posílání dat na maticové displeje, proto aby zobrazovací frekvence názvů lokality byla alespoň **100Hz**. Při zjišťování vhodného časového intervalu nastaly komplikace, protože minimální hodnota časového zpoždění byla 5ms a v aplikaci bylo požadováno zpoždění 1ms, proto bylo nutné přenastavit hodnotu *Task_Tick* v MQX.

Další komplikace vznikla u přenosu dat po sběrnici, kde často docházelo k špatnému poslání dat a přerušení spojení. Bylo to způsobeno tím, že sběrnice I2C1 byla zarušena ze spínaných zdrojů a také z proudových pulzů při spínání jednotlivých sloupců maticových displejů. To bylo vyřešeno stíněním datových vodičů, použitím filtračních kondenzátorů na zdrojích a modulech a omezením proudových pulzů. U jednotlivých modulů TimePanel_1 a TimePanel_3 byly také změněny hodnoty předřadných odporů displejů z důvodu rozdílné svítivosti modulů TimePanelu. Také bylo zapotřebí při inicializaci, v úloze *Main_Task*, nastavit vhodné časové zpoždění pro jednotlivé části úlohy. Například u připojení k síti, kde je nutné vyčkat určitou dobu, než se zařízení připojí anebo při testování segmentu a teček displejů, kde se nastaví doba svícení.



Obr. 79 TimePanel

Po doladění aplikace i HW, bylo provedeno kompletní testování TimePanelu, kdy byly testovány všechny režimy přepínání lokalit, všechny příkazy v příkazovém řádku přes sériovou linku (USB) i síť (Telnet Server) a připojení k síti pomocí dynamické i statické IP. Také byla provedena kontrola, zda je u lokalit nastaveno správné časové pásmo a zobrazen přesný čas s nastavením letního či zimního času. U kontroly změny letního a zimního času byl manuálně přenastavován čas, aby se zjistilo, zda ke změně času (posunu) dojde jen pro dané lokality v daný den a okamžik. TimePanel prošel kompletním testem úspěšně, viz fotografie (Obr. 79) funkčního zařízení.

6 Závěr

Úkolem této diplomové práce bylo navrhnout a zrealizovat zařízení pro zobrazení světového času – TimePanel. Pro realizaci práce bylo nutné se seznámit s požadavky ze strany zadavatele a stávajícími návrhy řešení. Dále bylo nutné nastudovat základní informace o časových zónách, způsobech získávání přesného času a použitých nástrojích jako byl *RTOS MQX* či *MCU PXN20* a Freescale Tower System. TimePanel je vnitřní závěsné zařízení umístěné na chodbě pod stropem. Je oboustranný, přehledný a dobře viditelný na velkou vzdálenost při různých světelných podmínkách. Na TimePanelu je zobrazen přesný čas, datum a název lokality pro sedm lokalit s rozdílnými časovými pásmy. Čas, datum a název lokality je zobrazen na modulech s maticovými a sedmi segmentovými displeji. Pro mezimodulovou komunikaci mezi jednotlivými moduly je použita sběrnice I2C. Přesný čas je synchronizován z časových serverů pomocí internetového připojení. Mezi jednotlivými lokalitami se lze přepínat pomocí tlačítka v konstrukci TimePanelu. Toto tlačítko slouží také k výběru a nastavení automatického nebo manuálního režimu přepínání lokalit. Návrh a realizaci tohoto zařízení jsem rozdělil do tří částí. V první, mechanické části, jsem se zaměřil na návrh a realizaci konstrukce. Návrhu a realizaci DPS modulu TimePanelu jsem se věnoval v druhé části. A v třetí, softwarové části jsem vytvořil a otestoval kompletní aplikaci pro zařízení s operačním systémem *RTOS MQX*.

Konstrukce oboustranného TimePanelu je hliníková a šroubovaná o rozměrech **720x 540x60mm**. Nosným prvkem konstrukce je rám tvořený čtyřmi hliníkovými profily, které se vzájemně sešroubují pomocí hliníkových konzol. V rámu je upevněna plastová středová deska oboustranně osazená LED moduly TimePanel 1 až 3, napájecími zdroji a řídicí jednotkou. Moduly TimePanel jsou na středovou desku uchyceny distančními sloupky. Díky sešroubování všech dílů se jedná o konstrukčně nejjednodušší řešení. TimePanel je zavěšen pomocí nerezového lanka protaženého malými dírkami v horním profilu rámu a ukotveno pomocí lanové svorky DIN741. Před moduly na obou stranách TimePanelu je čiré plexisklo s polepovou a protisluneční folii, přes které jde vidět pouze displeje modulu. Plexisklo je také použito z důvodu zvýšení odolnosti panelu proti vniknutí cizího tělesa či kapaliny a je uchyceno pomocí nedýmových magnetů.

TimePanel je napájen z elektrické sítě 230V a obsahuje spínané zdroje s výstupním napětím 12V, 5V a 3.3V. Je sestaven z tří různých typů modulů s maticovými a sedmi segmentovými LED displeji. Modul TimePanel 2 obsahuje dvě led diody a dva sedmi segmentové displeje, je největší a slouží k zobrazení hodin. Modul TimePanel 3 obsahuje dva sedmi segmentové displeje, je nejmenší a slouží k zobrazení data. Modul TimePanel 1 obsahuje jeden sedmi segmentový displej a slouží k zobrazení názvu lokality – jednoho znaku. V zařízení je dohromady použito dvacet dva modulu TimePanel 1 až 3. Ve všech těchto modulech jsou použity společné součástky jako např. expandér MCP23016, tranzistorové pole atd. Tyto moduly mají všechny stejnou strukturu rozmístění konektorů a displejů pro snadné propojení navzájem. V aplikaci jsou použity dvě sběrnice I2C pro komunikaci mezi moduly a řídicí jednotkou. Řídicí jednotka komunikuje s moduly TimePanel 2 a 3 přes sběrnici I2C0 a s moduly TimePanel 1 pomocí sběrnice I2C1. Jako řídicí jednotka je použit Tower *PC- PXN20* s 32 bitovým procesorem (*dual core*) od firmy FreeScale.

Aplikace pro TimePanel je napsána pro operační systém *RTOS MQX* v jazyce C. Operační systém je velmi stabilní a jeho použití umožnilo rozdělení aplikace do úloh a procesů. Při inicializaci systému jsou na TimePanelu postupně otestovány segmenty všech displejů, ověří se, zda nastavení *DST* a synchronizace času proběhla úspěšně a na displejích se zobrazí IP adresa zařízení, potřebná pro

Telnet. Poté jsou vytvořeny úlohy pro zobrazení názvu lokality, přesného času a data na TimePanelu. Při zobrazování názvu lokality je důležité, aby obnovovací frekvence byla 100Hz. Zařízení získává přesný čas z Time Serveru (*SNT*P) periodicky, co 2 minuty. Zařízení je možné nastavit a komunikovat s ním (za běhu), pomocí příkazového řádku (*konzole*) přes sériovou linku (*USB*) nebo síťové připojení (*Telnet Server*). Pro nastavení TimePanelu z konzole jsou v aplikaci vytvořené příkazy pro manuální nastavení času, data, volbu režimu přepínání lokalit atd. U tlačítka použitého pro přepínání režimu a lokalit se rozlišují dva typy stisku tlačítka (*‚Krátký‘* a *‚Dlouhý‘*). Při *‚krátkém‘* stisku dojde k manuálnímu přepnutí lokality a v případě *‚dlouhého‘* stisku se nastaví jeden z dvou automatických režimů přepínání lokalit. Celá aplikace je popsána UML diagramy.

Po sestavení konstrukce spolu s moduly, řídicí jednotkou a zdroji, bylo provedeno kompletní testování TimePanelu, kdy byly testovány všechny režimy přepínání lokalit, všechny příkazy v příkazovém řádku přes sériovou linku (*USB*) i síť (*Telnet Server*) a připojení k síti pomocí dynamické i statické IP. Také byla provedena kontrola, zda je u lokalit nastaveno správné časové pásmo a je zobrazen přesný čas s nastavením letního či zimního času. TimePanel prošel kompletním testem úspěšně.

Na závěr diplomové práce bych chtěl dodat, že jsem splnil všechny body zadání a stanovené cíle a TimePanel je funkční. Na práci jsem průběžně pracoval již od února minulého roku. V průběhu práce jsem získal nové cenné poznatky.

7 Použitá literatura

- [1] HORSKÝ, Petr. *Formica 4.2 : návrhový systém pro plošné spoje : editor plošného spoje a autorouter: příručka uživatele* . 1. Praha : BEN, 1998. 190 s. ISBN 80-86056-34-1.
- [2] KŘIVKA, Ivo. *Formica 4.2 : návrhový systém pro plošné spoje : schematický editor : příručka uživatele*. 1. Praha : BEN, 1998. 248 s. ISBN 80-86056-35-X.
- [3] CIRKOVSKÝ, Jaroslav. *Metody návrhů desek plošných spojů* [online]. Brno, 2008. 53 s. Bakalářská práce. VUT Brno. Dostupné z WWW: <http://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=7581>.
- [4] SROVNAL, Vilém. *Operační systémy pro řízení v reálném čase*. Ostrava : VŠB-TU Ostrava, 2003. 212 s.
- [5] HW [online]. 2005 [cit. 2010-10-27]. Dostupné z WWW: <<http://hw.cz/>>.
- [6] Root [online]. 2009 [cit. 2010-12-28]. Komunikace po sériové sběrnici I2C. Dostupné z WWW: <<http://www.root.cz/clanky/komunikace-po-seriove-sbernici-isup2supc/>>.
- [7] Freescale Semiconductors [online]. c2009 [cit. 2009-11-11]. Dostupný z WWW: <<http://www.freescale.com/>>.
- [8] FreeScale TWR-PXN20 [online]. c2012 [cit. 2012-11-01]. Dostupný z WWW: <http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=TWR-PXN20>.
- [9] KAKÁČ, Tomáš. *Elektronický informační panel*. Brno, 2010. Bakalářská práce. VUT Brno
- [10] VIRIUS, Miroslav. *Jazyky C a C++ : Kompletní kapesní průvodce programátora*. první vydání. Praha : Grada Publishing, a.s., 2006. 520 s. ISBN 80-247-1494-9.
- [11] Freescale MQX™ Real-Time Operating System (RTOS). [online]. [cit. 2012-11-01]. Dostupné z: <http://www.freescale.com/webapp/sps/site/overview.jsp?code=MQXRTOS>
- [12] MCP23016: 16-Bit I2C™ I/O Expandér. [online]. [cit. 2012-11-20]. Dostupné z: <http://ww1.microchip.com/downloads/en/DeviceDoc/20090C.pdf>
- [13] ING. JIŘÍ KOTZIAN, Ph.D. *Navrhování a realizace vestavěných řídicích systémů: učební text a návody do cvičení*. Ostrava: VŠB – TUO, 2012.
- [14] Tower Serial Module: User Manual for TWR-SER. <Http://www.freescale.com> [online]. 2012 [cit. 2013-01-18]. Dostupné z: http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=TWR-SER

- [15] MOŽNÝ, Karel. *Implementace ovladače I2S Audio v systému Freescale MQX RTOS*. Brno, 2012. Diplomová práce. VUT Brno.
- [16] Co umožňuje a programátorovi přináší RTOS MQX od Freescale ?. In: VOJÁČEK, Antonín. *www.HW.cz* [online]. 2011 [cit. 2013-01-18]. Dostupné z: <http://www.hw.cz/novinky/freescale/co-umoznuje-a-programatorovi-prinasi-rtos-mqx-od-freescale-1dil.html>
- [17] Sběrnice I2C: o co jde a jak pracuje. *DH servis* [online]. 2008 [cit. 2013-02-01]. Dostupné z: <http://www.dhservis.cz/iic.htm>
- [18] Přesný čas na PC prostřednictvím Internetu. In: POUPA, Martin. *Http://www.hw.cz/* [online]. 2004 [cit. 2013-02-04]. Dostupné z: <http://www.hw.cz/teorie-a-praxe/dokumentace/presny-cas-na-pc-prostrednictvim-internetu.html>
- [19] *MobaTime* [online]. 1995 [cit. 2013-02-04]. Dostupné z: <http://www.mobatime.cz/>
- [20] Externí sériové sběrnice SPI a I²C. In: *Root.cz* [online]. 2008 [cit. 2013-04-23]. Dostupné z: <http://www.root.cz/clanky/externi-seriove-sbernice-spi-a-i2c/>
- [21] Letní čas. In: *Wikipedia: the free encyclopedia* [online]. 2012 [cit. 2013-04-23]. Dostupné z: http://cs.wikipedia.org/wiki/Letn%C3%AD_%C4%8Das
- [22] ICalendar. In: *Wikipedia: the free encyclopedia* [online]. 2011 [cit. 2013-04-23]. Dostupné z: <http://cs.wikipedia.org/wiki/ICalendar>

Seznam příloh

- Příloha I.** Fotografie TimePanelu a jeho součástí
- Příloha II.** Modul TimePanel_1
- Příloha III.** Modul TimePanel_2
- Příloha IV.** Modul TimePanel_3
- Příloha V.** Rozšiřující deska ‚*Expansion_BOARD*‘
- Příloha VI.** Soupis materiálů pro konstrukci TimePanelu
- Příloha VII.** UML diagramy aplikace TimePanel

Přiložené DVD dále obsahuje:

1. Diplomová_práce_LIP130.docx – vlastní práce
2. Diplomová _práce_LIP130.pdf – vlastní práce
3. /OBR/UML - Složka, ve které jsou vlastní UML použité v diplomové práci
4. /OBR/SCHEMATA - Složka, ve které jsou vlastní obrázky elektrických a blokových schémat
5. /OBR/OBR_CIT - Složka, ve které jsou citované obrázky
6. /OBR/OBR_VLASTNI- Složka, ve které jsou vlastní obrázky z diplomové práce
7. /OBR/ZADANI - Složka, ve které je zadání diplomové práce
8. /FOTO - Složka, ve které je umístěna fotodokumentace
9. /PDF/MQX - Složka ve které jsou umístěny katalogové listy RTOS MQX
10. /PDF/PXN20 - Složka ve které jsou umístěny katalogové listy PXN20
11. /PDF/PARTS - Složka ve které jsou umístěny katalogové listy použitých součástek
12. /PDF/VYKRESY- Složka ve které je umístěna vlastní výkresová dokumentace a 3D modely konstrukce

Příloha I. Fotografie TimePanelu a jeho součástí

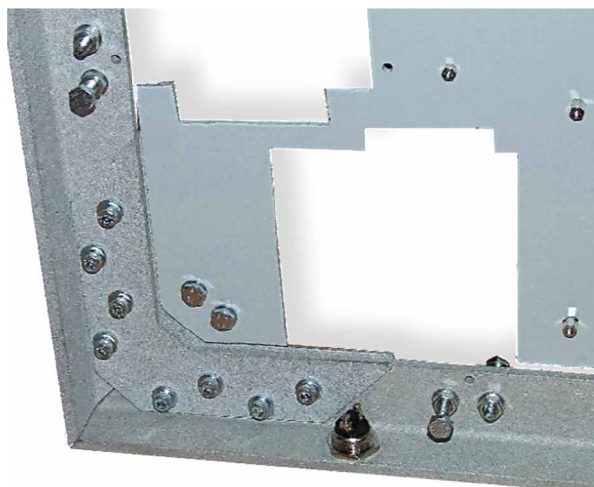


Figure 1 Detail konstrukce se středovou deskou a tlačítkem

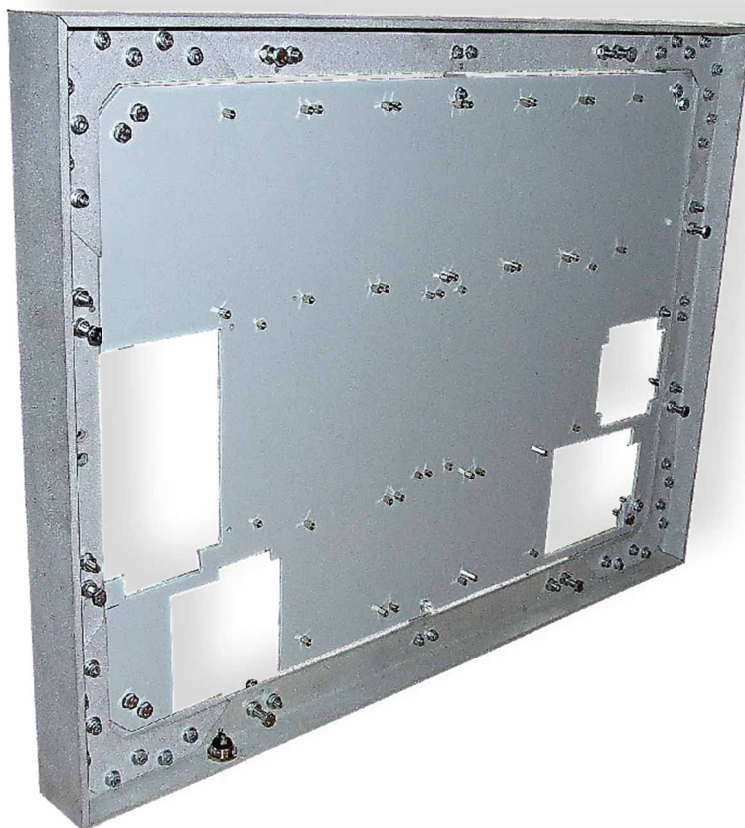


Figure 2 Konstrukce TimePanelu

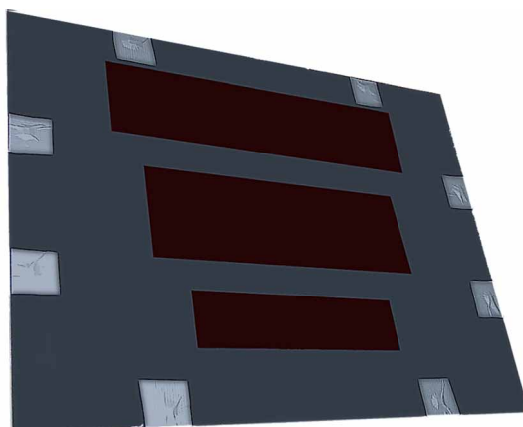


Figure 3 Plexisklo TimePanelu - pohled ze strany modulu

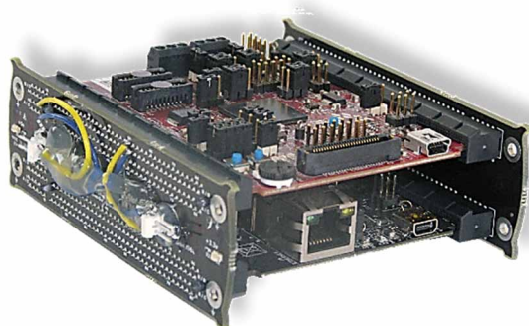


Figure 4 Freescale Tower modul PXN20 v mini elevátorech



Figure 5 Fotografie TimePanelu



Figure 6 Otevřený TimePanel při testování s PXN20 v čtyř-slotových elevátorech



Figure 7 Fotografie TimePanelu– čelní pohled



Figure 8 Poměr velikosti TimePanelu s postavou



Figure 9 Fotografie TimePanelu upravená v aplikaci GIMP

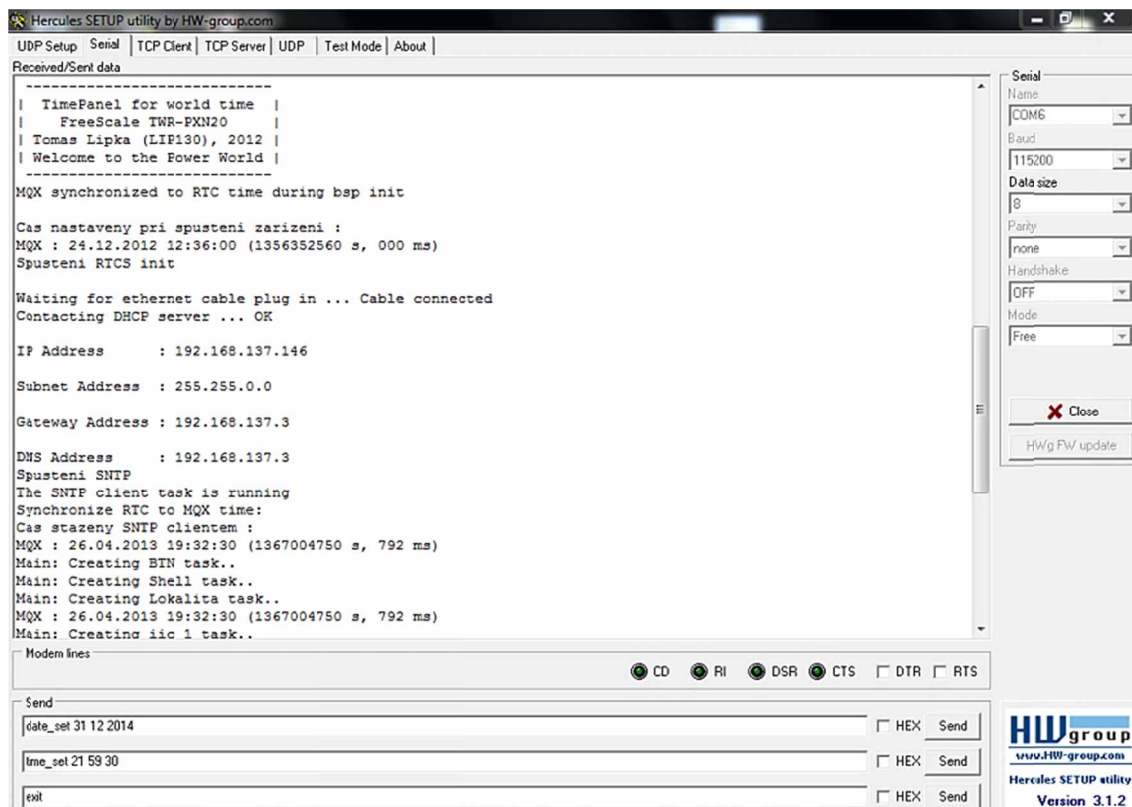


Figure 10a Ukázka spuštění TimePanelu v konzoli přes USB

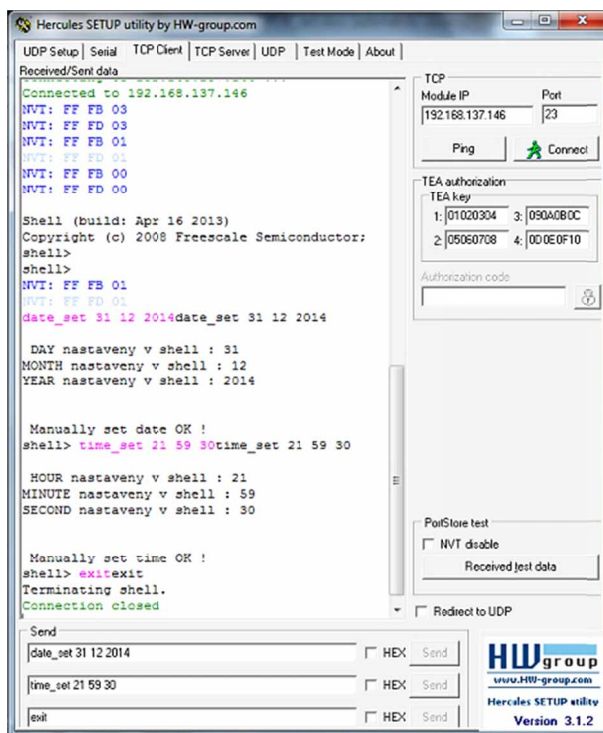


Figure 6 Ukázka manuálního nastavení času a data TimePanelu v konzoli přes síť (Telnet)

Příloha II. Modul TimePanel_1

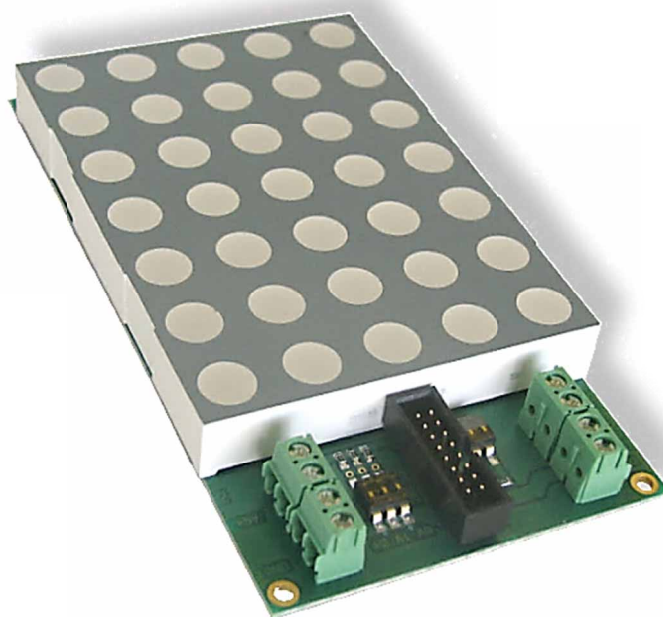


Figure 7 Modul TimePanel_1

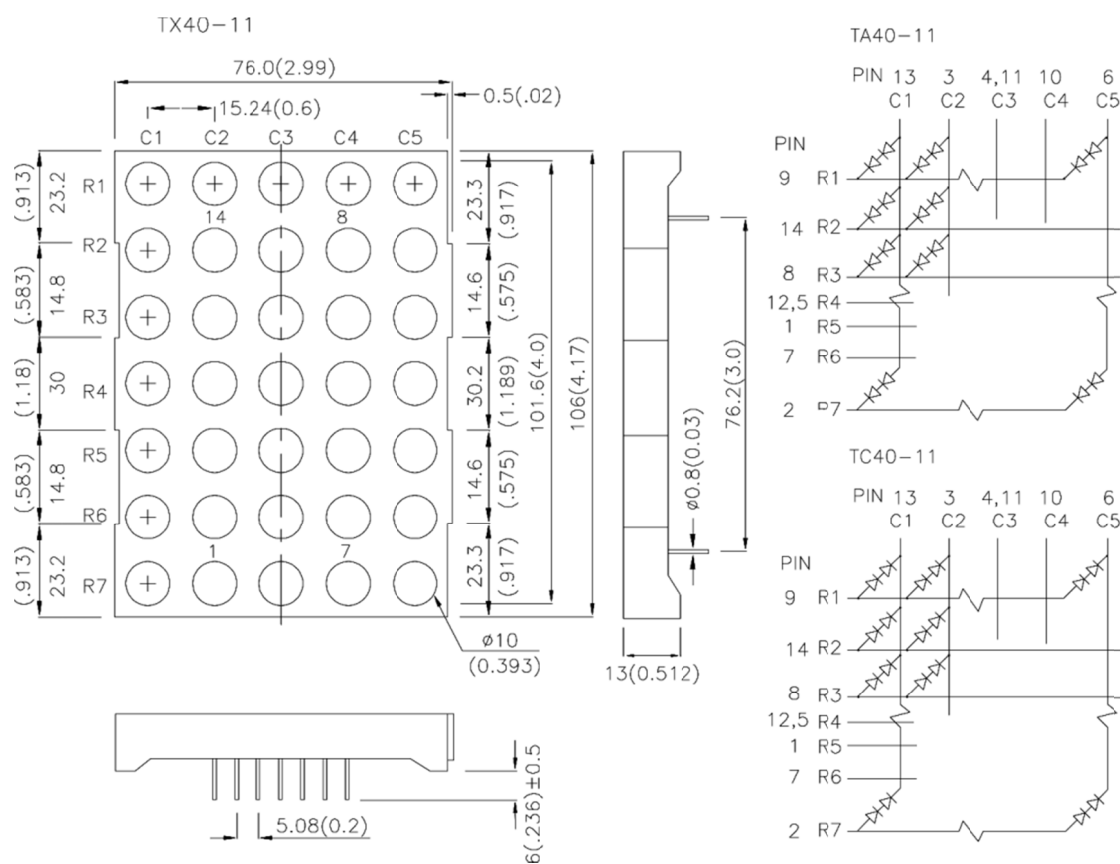


Figure 8 Rozměry a zapojení 5x7 maticového displeje TA40-11SRWA

Parametr	Hodnota	Jednotka
Společný vývod	anoda	[-]
Počet znaků	5x7	[-]
Barva	Super Bright Red	[-]
Vlnová délka	660	[nm]
Napětí v propustném směru (typ.)	1.85	[V]
Max. napětí v propustném směru	2.5	[V]
Proud v propustném směru (typ.)	20	[mA]
Max. proud v propustném směru	30	[mA]
Velikost displeje	106x76	[mm]
Zapojení číslic	samostatné vývody	[-]
Svítivost	44	[mcd]

Figure 9 Technické parametry maticového LED displeje TA40-11SRWA

Označení	Množství	Výrobce	Typové označení	Vlastnosti	Provedení	Prodějna	Objednací kód	Cena za kus (Kč)	Cena za množství (Kč)
DI1	1	KINGBRIGHT ELECTRONIC	TA40-11SRWA	LED; matice; 5x7; červená; 18-44mcd; anoda	THT	TME.EU		160	160
R2-R7,R39	7	MULTICOMP	MCPWR10FTEQ1300	RESISTOR, 2010, 130R 1%, 0.75W	SMD 2010	Farnell.com	1887596RL	3.6	25.2
Q2	1	TEXAS INSTRUMENTS	ULN2803ADW	TRANSISTOR ARRAY,8NPN, 50V,18-SOIC	SMD 18-SOIC	Farnell.com	1652419	26	26
R22-R28	7	MULTICOMP	MCMR08X1003FTL	RESISTOR, 0805, 100K, 1%, ANTI SULFUR	SMD 0805	Farnell.com	2073608	0.292	2.044
IO1	1	MICROCHIP	MCP23016-I/SO	IC, IO EXPANDER 16BIT, SMD, SOIC28	SMD SOIC28	Farnell.com	1439758	43	43
R29,R30,R31,R8,R9,R11,R14,R15,R18,R19,R37,R38,R42	14	PANASONIC	ERJ6GEY103V	RESISTOR, 0805, 10K 5%, 0.125W	SMD 0805	Farnell.com	2057719RL	0.525	7.35
R34,R35,R10,R13,R17,R21,R41	7	VISHAY DRALORIC	CRCW08050000Z0EA	RESISTOR, 0805, 0R0	SMD 0805	Farnell.com	1469846	0.422	2.954
R36	1	VISHAY DRALORIC	CRCW08053K90FKEAHP	RESISTOR, 0805, 1%, 3K90	SMD 0805	Farnell.com	1738967RL	1.5	1.5
R33,R32	2	MULTICOMP	MCPWR05FTEW4701	RESISTOR, 0805, 4.7K 1%, 0.125W	SMD 0805	Farnell.com	1887298	1.27	2.54
C1	1	MULTICOMP	A0805C330JCT	CAPACITOR, 0805, 33PF, 100V	SMD 0805	Farnell.com	9406255	1.53	1.53
SW1	1	MULTICOMP	MCEMR-03-T	SWITCH, DIL, SMD, 3WAY	SMD	Farnell.com	1524006	18	18
SW2	1	MULTICOMP	MCEMR-02-T	SWITCH, DIL, SMD, 2WAY	SMD	Farnell.com	1524005	17	17
H2	1	MOLEX	70246-1604	HEADER, SHROUDED, 16WAY	THT	Farnell.com	1392536	27	27
K1-K4	4	PHOENIX CONTACT	PT1,5/2-5.0-H	TERM BLK, PCB, SCREW, 5.0MM, 2WAY	THT	Farnell.com	1793674	8.5	34
R1,R12,R16,R20,R40	5	MULTICOMP	MCSR08X1001FTL	RESISTOR, 0805, 1K, 1%, ANTI	SMD 0805	Farnell.com	2074332	0.3	1.5
T1,T3,T5,T7,T9	5	MULTICOMP	BC847B	TRANSISTOR, NPN, SOT-23	SMD SOT-23	Farnell.com	1574343RL	1.27	6.35
T2,T4,T6,T8,T10	5	RECTIFIER INTERNATIONAL	IRF7416TRPBF	P CHANNEL MOSFET, 10A	SMD SO8	Farnell.com	2119119	15	75
C2	1	MURATA	GRM219F51E105ZA01D	CAPACITOR, 0805, Y5V, 25V, 1UF	SMD 0805	Farnell.com	1828853	1.1	1.1
C3	1	KEMET	T491C106K016AT	CAPACITOR, CASE C, 10UF, 16V	SMD C (6032)	Farnell.com	1457461	8	8
Cena celkem pro 1 kus modulu TimePanelu_1									460.068

Figure 10 Part list

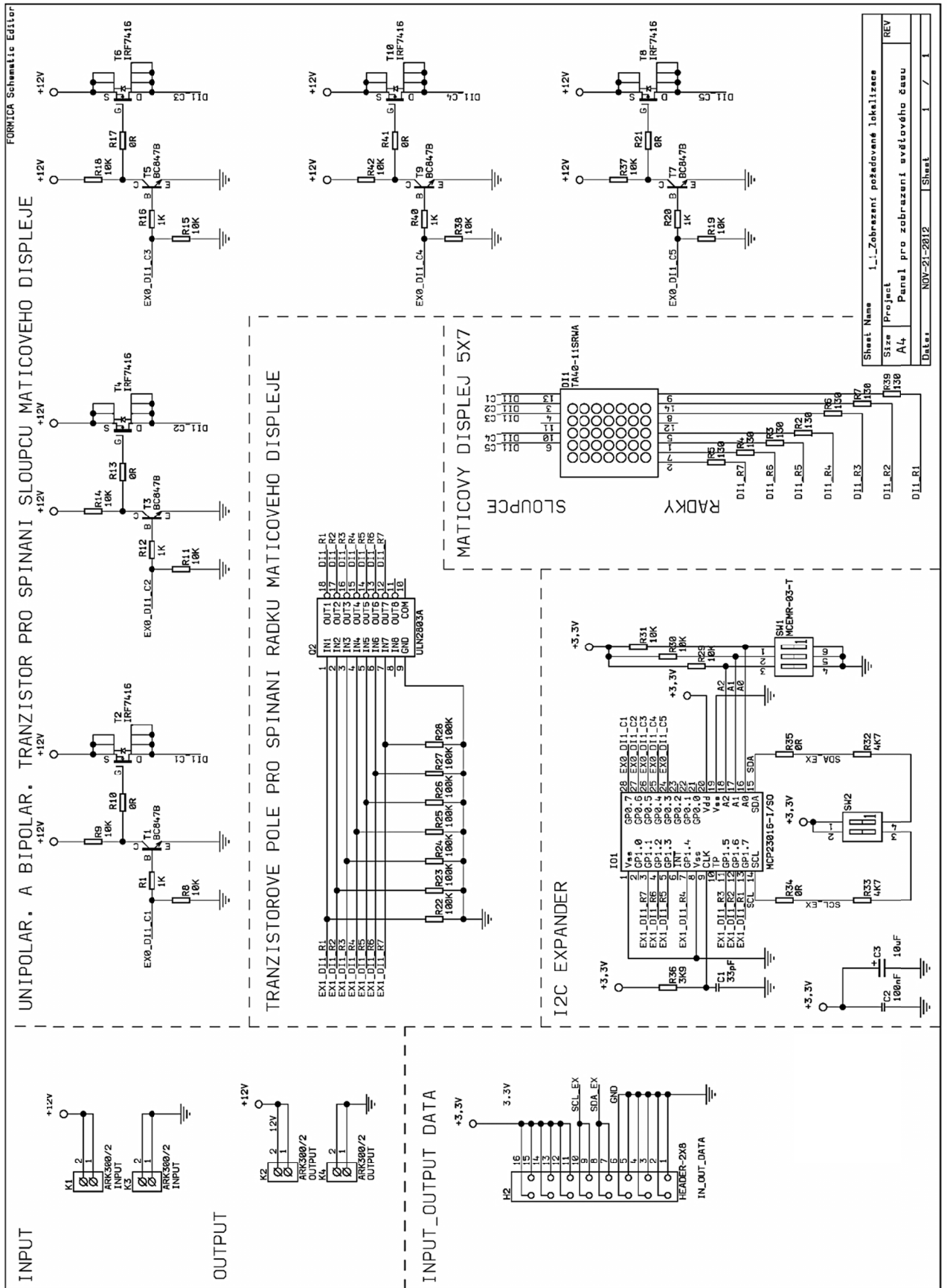


Figure 11 Schéma zapojení modulu TimePanel_1

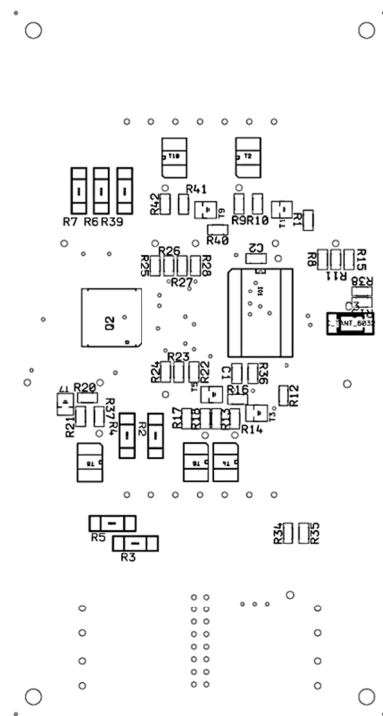
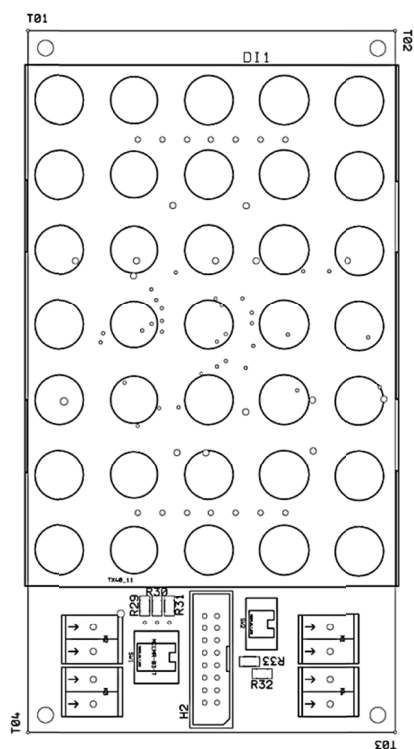


Figure 12 Osazovací plán DPS (TOP a BOTTOM) pro TimePanel_1

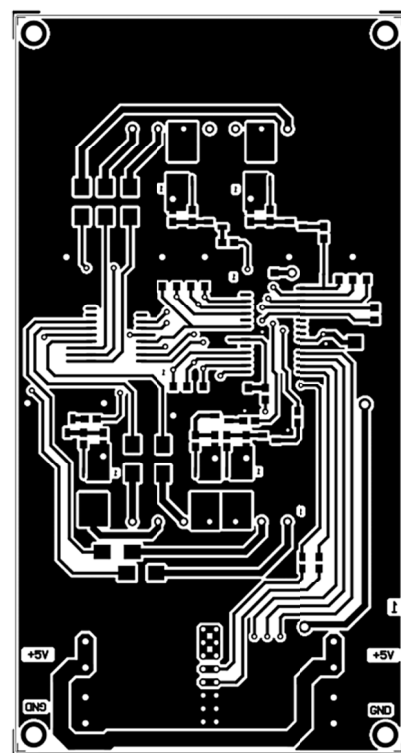
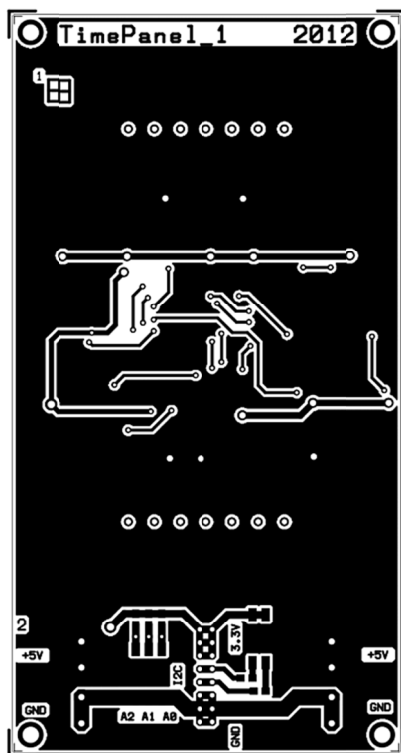


Figure 13 Vrstvy spojů (TOP a BOTTOM) pro TimePanel_1

Příloha III. Modul TimePanel_2

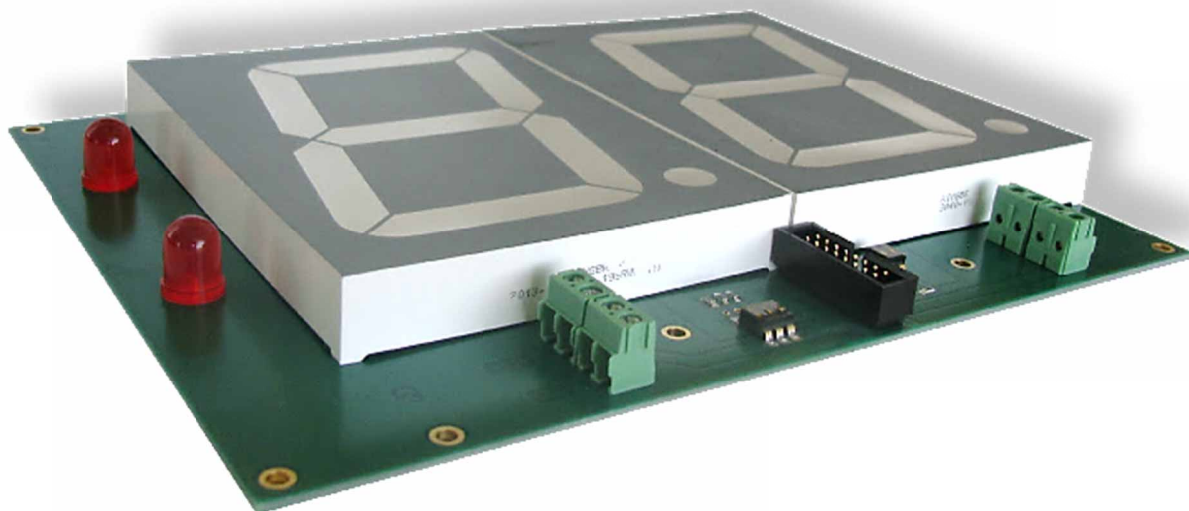


Figure 14 Modul TimePanel_2

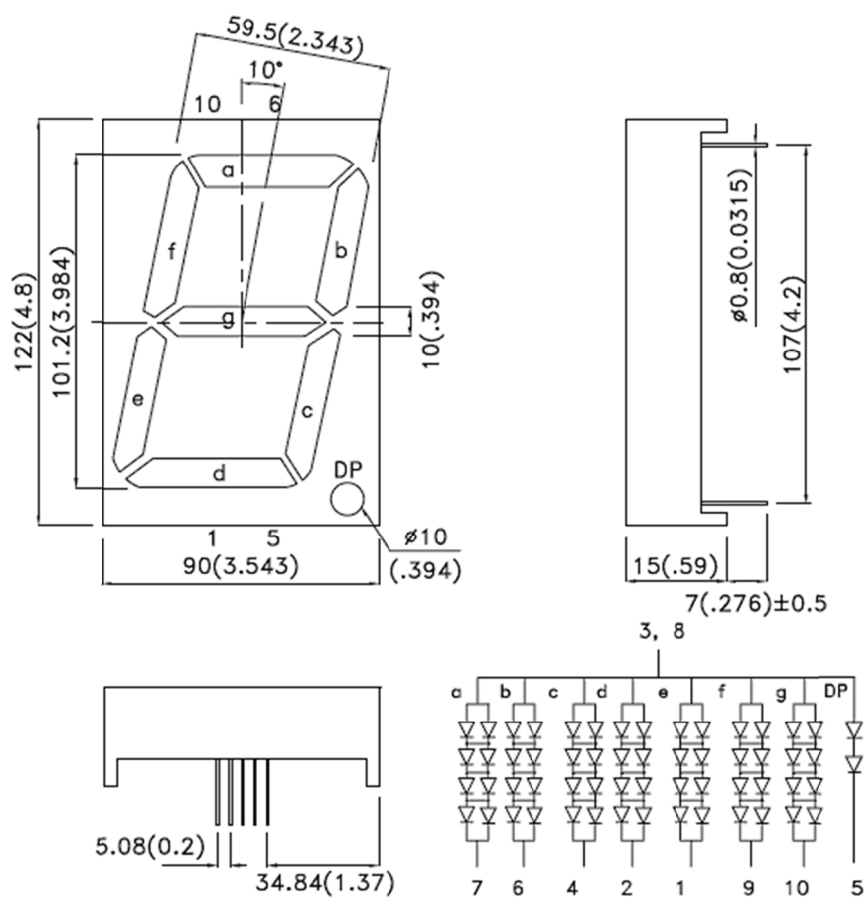


Figure 15 Rozměry a zapojení sedmi segmentového displeje SA40-19SRWA

Parametr	Hodnota	Jednotka
Společný vývod	anoda	[-]
Počet znaků	1	[-]
Desetinná tečka	vpravo	[-]
Barva	Super Bright Red	[-]
Vlnová délka	660	[nm]
Napětí v propustném směru (typ.)	7.4	[V]
Max. napětí v propustném směru	10	[V]
Proud v propustném směru (typ.)	20	[mA]
Max. proud v propustném směru	60(30)	[mA]
Výška znaku	101	[mm]
Zapojení číslic	samostatné vývody	[-]
Svítivost	105	[mcd]

Figure 16 10.1.1.1. Technické parametry LED displeje SA40-19SRWA

Parametr	Hodnota	Jednotka
Vyzařovací uhel	60	[°]
Barva	HIGH EFFICIENCY RED	[-]
Vlnová délka	660	[nm]
Napětí v propustném směru (typ.)	2	[V]
Max. napětí v propustném směru	2.5	[V]
Proud v propustném směru (typ.)	20	[mA]
Max. proud v propustném směru	30	[mA]
Průměr diody	10	[mm]
Zapojení	samostatné vývody	[-]
Svítivost	100	[mcd]

Figure 17 Technické parametry LED diody KINGBRIGHT - L-813ID

Označení	Množství	Výrobce	Typové označení	Vlastnosti	Provedení	Prodávající	Objednací kód	Cena za kus (Kč)	Cena za množství (Kč)
D1, D2	2	KINGBRIGHT ELECTRONIC	SA40-19SRWA	LED; jednomístný 7-segmentový; 101,2mm; červená, anoda	THT	TME.EU		185	370
R1-R14	14	MULTICOMP	MCPWR10FTEQ1500	RESISTOR, 2010, 150R 1%, 0.75W	SMD 2010	Farnell.com	1887597RL	3.6	50.4
Q1, Q2	2	TEXAS INSTRUMENTS	ULN2803ADW	TRANSISTOR ARRAY, 8NPN, 50V, 18-SOIC	SMD 18-SOIC	Farnell.com	1652419	26	52
R15-28, R37	15	MULTICOMP	MCMR08X1003FTL	RESISTOR, 0805, 100K, 1%, ANTI SULFUR	SMD 0805	Farnell.com	2073608	0.292	4.38
IO1	1	MICROCHIP	MCP23016-I/SO	IC, IO EXPANDER 16BIT, SMD, SOIC28	SMD SOIC28	Farnell.com	1439758	43	43
R29, R30, R31	3	PANASONIC	ERJ6GEY1103V	RESISTOR, 0805, 10K 5%, 0.125W	SMD 0805	Farnell.com	2057719RL	0.525	1.575
R34, R35	2	VISHAY DRALORIC	CRCW0805000020EA	RESISTOR, 0805, 0R0	SMD 0805	Farnell.com	1469846	0.422	0.844
R36	1	VISHAY DRALORIC	CRCW08053K90FKEAHP	RESISTOR, 0805, 1%, 3K90	SMD 0805	Farnell.com	1738967RL	1.5	1.5
R33, R32	2	MULTICOMP	MCPWR05FTEW4701	RESISTOR, 0805, 4.7K 1%, 0.125W	SMD 0805	Farnell.com	1887298	1.27	2.54
C1	1	MULTICOMP	A0805C330JCT	CAPACITOR, 0805, 33PF, 100V	SMD 0805	Farnell.com	9406255	1.53	1.53
SW1	1	MULTICOMP	MCEMR-03-T	SWITCH, DIL, SMD, 3WAY	SMD	Farnell.com	1524006	18	18
SW2	1	MULTICOMP	MCEMR-02-T	SWITCH, DIL, SMD, 2WAY	SMD	Farnell.com	1524005	17	17
D1, D2	2	KINGBRIGHT	L-813ID	LED, 10MM, HE-RED	THT	Farnell.com	1168543	5.1	10.2
R38	1	MULTICOMP	MCPWR10FTEQ3600	RESISTOR, 2010, 360R 1%, 0.75W	SMD 2010	Farnell.com	1887606	3.6	3.6
H1	1	MOLEX	70246-1604	HEADER, SHROUDED, 16WAY	THT	Farnell.com	1392536	27	27
K1-K4	4	PHOENIX CONTACT	PT1,5/2-5.0-H	TERM BLK, PCB, SCREW, 5.0MM, 2WAY	THT	Farnell.com	1793674	8.5	34
C2	1	MURATA	GRM219F51E105ZA01D	CAPACITOR, 0805, Y5V, 25V, 10F	SMD 0805	Farnell.com	1828853	1.1	1.1
C3	1	KEMET	T491C106K016AT	CAPACITOR, CASE C, 10UF, 16V	SMD C (6032)	Farnell.com	1457461	8	8
Celková cena za 1 kus TimePanelu 2 (Kč)									646.669

Figure 18 Part List

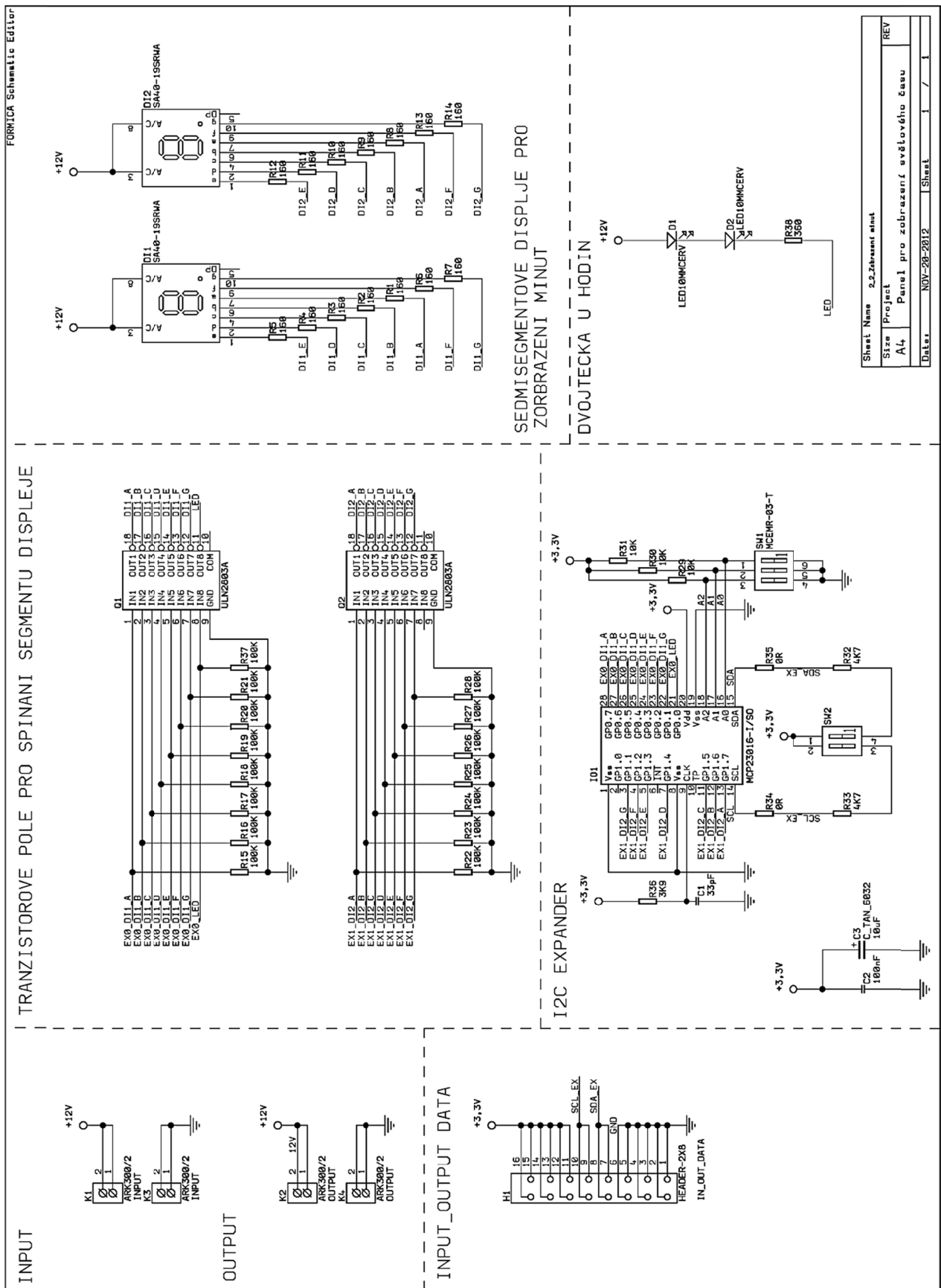


Figure 19 Schéma zapojení modulu TimePanel_2

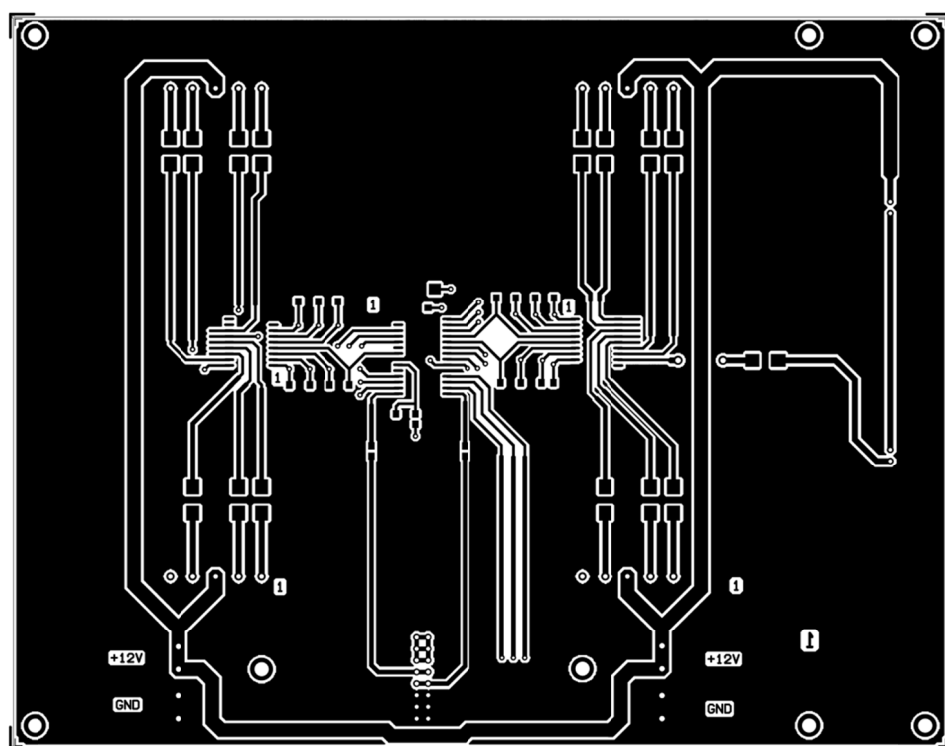
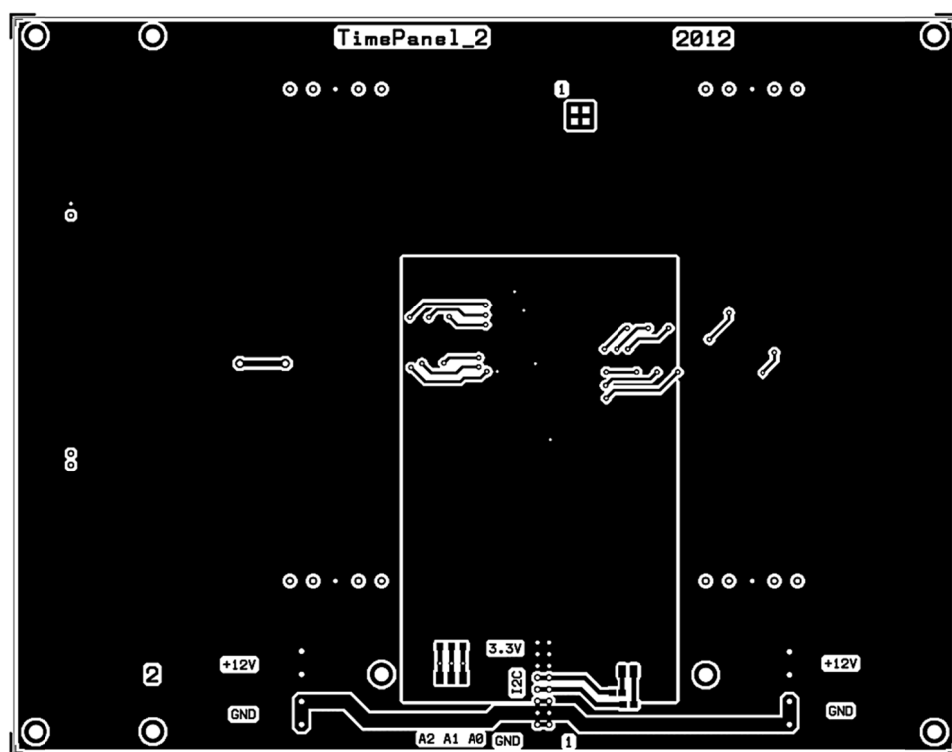


Figure 21 Vrstvy spojů DPS (TOP a BOTTOM) pro TimePanel_2

Příloha IV. Modul TimePanel_3

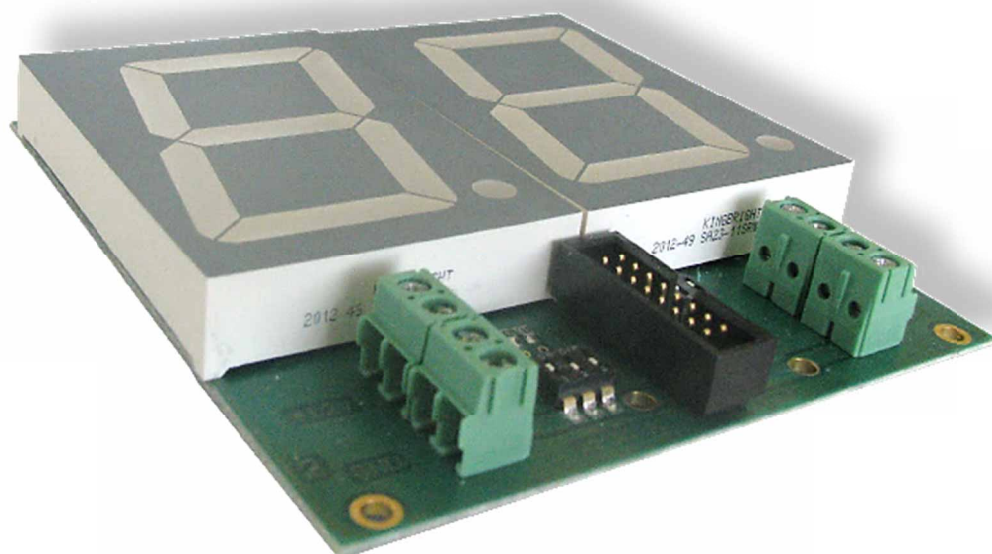


Figure 22 Modul TimePanel 3

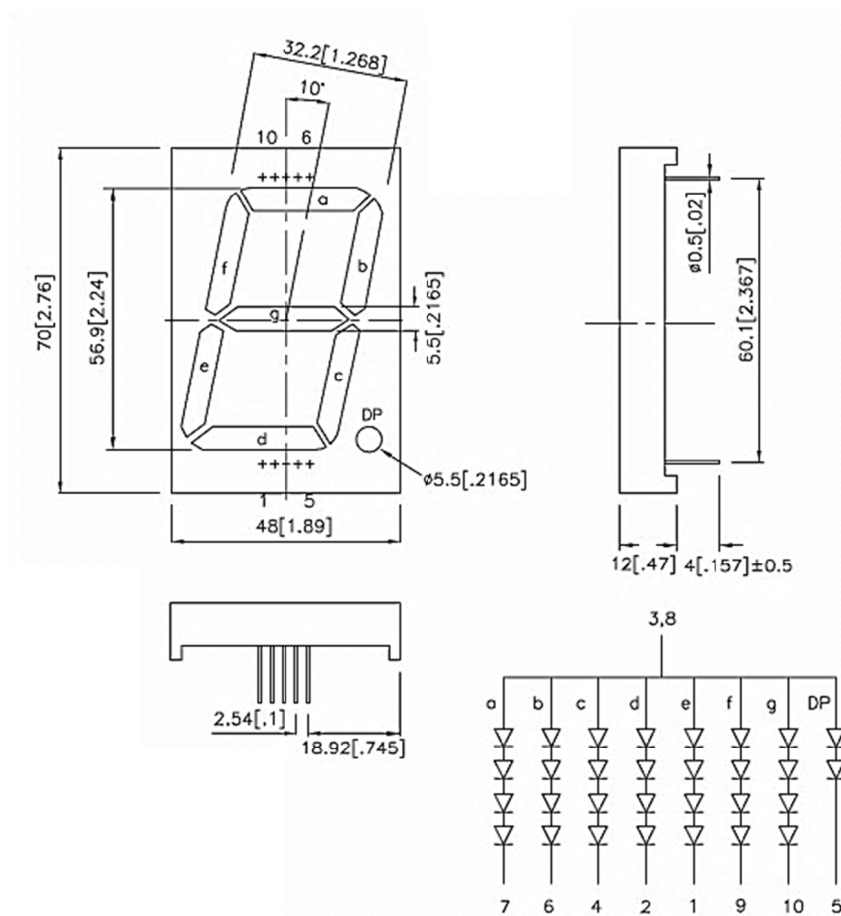


Figure 23 Rozměry a zapojení sedmi segmentového displeje SA23-12SRWA

Parametr	Hodnota	Jednotka
Společný vývod	anoda	[-]
Počet znaků	1	[-]
Desetinná tečka	vpravo	[-]
Barva	Super Bright Red	[-]
Vlnová délka	660	[nm]
Napětí v propustném směru (typ.)	7.4 (3.7)	[V]
Max. napětí v propustném směru	10 (5)	[V]
Proud v propustném směru (typ.)	20	[mA]
Max. proud v propustném směru	30	[mA]
Výška znaku	56.9	[mm]
Zapojení číslic	samostatné vývody	[-]
Svitivost	75	[mcd]

Figure 24 Technické parametry LED displeje SA23-12SRWA

Označení	Množství	Výrobce	Typové označení	Vlastnosti	Provedení	Prodávna	Objednací kód	Cena za kus (Kč)	Cena za množství (Kč)
D11, D12	2	KINGBRIGHT ELECTRONIC	SA23-11SRWA	LED; jednomístný 7-segmentový; 56,9mm; červená; anoda	THT	TME.EU		76	152
R1-R14	14	MULTICOMP	MCPWR10FTEQ1800	RESISTOR, 2010, 180R 1%, 0.75W	SMD 2010	Farnell.com	1887599RL	3.54	49.56
Q1, Q2	2	TEXAS INSTRUMENTS	ULN2803ADW	TRANSISTOR ARRAY, 8NPN, 50V, 18-SOIC	SMD 18-SOIC	Farnell.com	1652419	26	52
R15-28, R37, R38	16	MULTICOMP	MCMR08X1003FTL	RESISTOR, 0805, 100K, 1%, ANTI SULFUR	SMD 0805	Farnell.com	2073608	0.292	4.672
IO1	1	MICROCHIP	MCP23016-I/SO	IC, IO EXPANDER 16BIT, SMD, SOIC28	SMD SOIC28	Farnell.com	1439758	43	43
R29, R30, R31	3	PANASONIC	ERJ6GEYJ103V	RESISTOR, 0805, 10K 5%, 0.125W	SMD 0805	Farnell.com	2057719RL	0.525	1.575
R34, R35, R41, R42	4	VISHAY DRALORIC	CRCW08050000Z0EA	RESISTOR, 0805, 0R0	SMD 0805	Farnell.com	1469846	0.422	1.688
R36	1	VISHAY DRALORIC	CRCW08053K90FKEAHP	RESISTOR, 0805, 1%, 3K90	SMD 0805	Farnell.com	1738967RL	1.5	1.5
R33, R32	2	MULTICOMP	MCPWR05FTEW4701	RESISTOR, 0805, 4.7K 1%, 0.125W	SMD 0805	Farnell.com	1887298	1.27	2.54
C1	1	MULTICOMP	A0805C330JCT	CAPACITOR, 0805, 33PF, 100V	SMD 0805	Farnell.com	9406255	1.53	1.53
SW1	1	MULTICOMP	MCEMR-03-T	SWITCH, DIL, SMD, 3WAY	SMD	Farnell.com	1524006	18	18
SW2	1	MULTICOMP	MCEMR-02-T	SWITCH, DIL, SMD, 2WAY	SMD	Farnell.com	1524005	17	17
R39, R40	2	MULTICOMP	MCPWR10FTEQ3900	RESISTOR, 2010, 390R 1%, 0.75W	SMD 2010	Farnell.com	1887607RL	3.6	7.2
H1	1	MOLEX	70246-1604	HEADER, SHROUDED, 16WAY	THT	Farnell.com	1392536	27	27
K1-K4	4	PHOENIX CONTACT	PT1,5/2-5.0-H	TERM BLK, PCB, SCREW, 5.0MM, 2WAY	THT	Farnell.com	1793674	8.5	34
C2	1	MURATA	GRM219F51E105ZA01D	CAPACITOR, 0805, Y5V, 25V, 1UF	SMD 0805	Farnell.com	1828853	1.1	1.1
C3	1	KEMET	T491C106K016AT	CAPACITOR, CASE C, 10UF, 16V	SMD C (6032)	Farnell.com	1457461	8	8
Celková cena za 1 kus TimePanelu 3 (Kč)									413.265

Figure 25 Part List

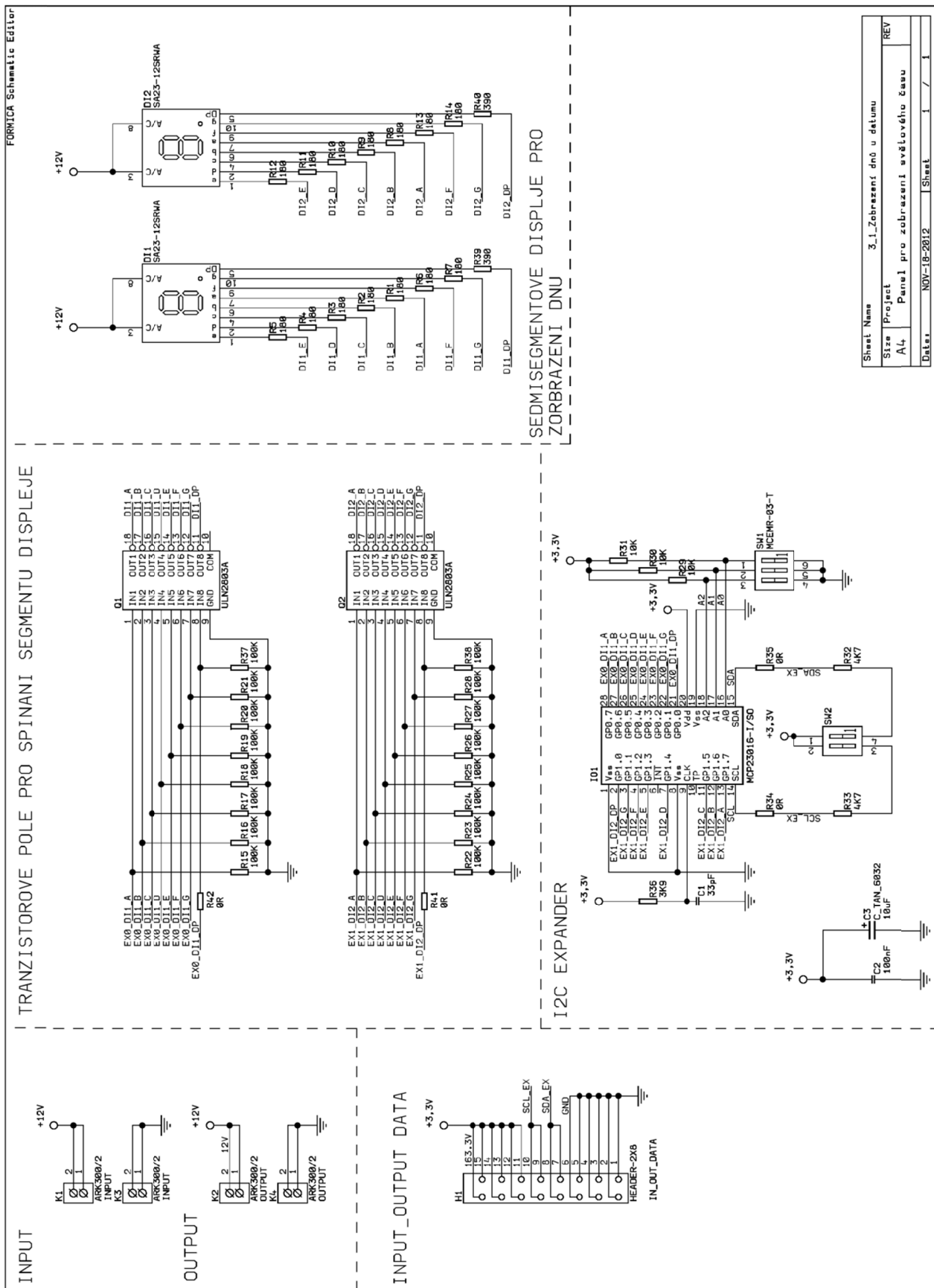


Figure 26 Schéma zapojení modulu TimePanel_3

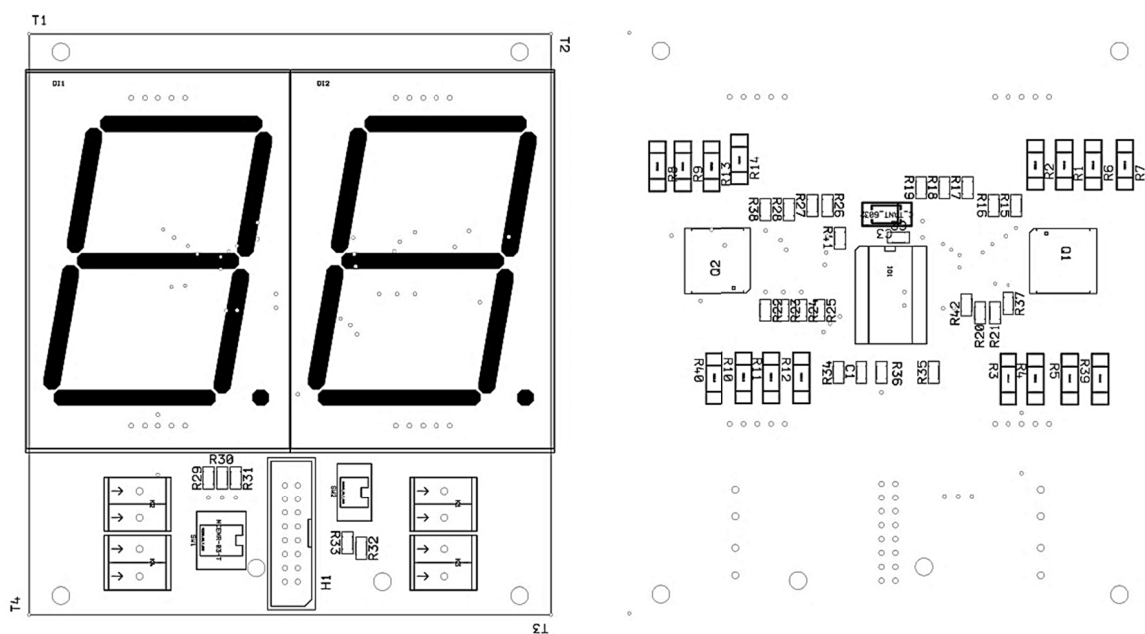


Figure 27 Osazovací plán DPS (TOP a BOTTOM) pro TimePanel_3

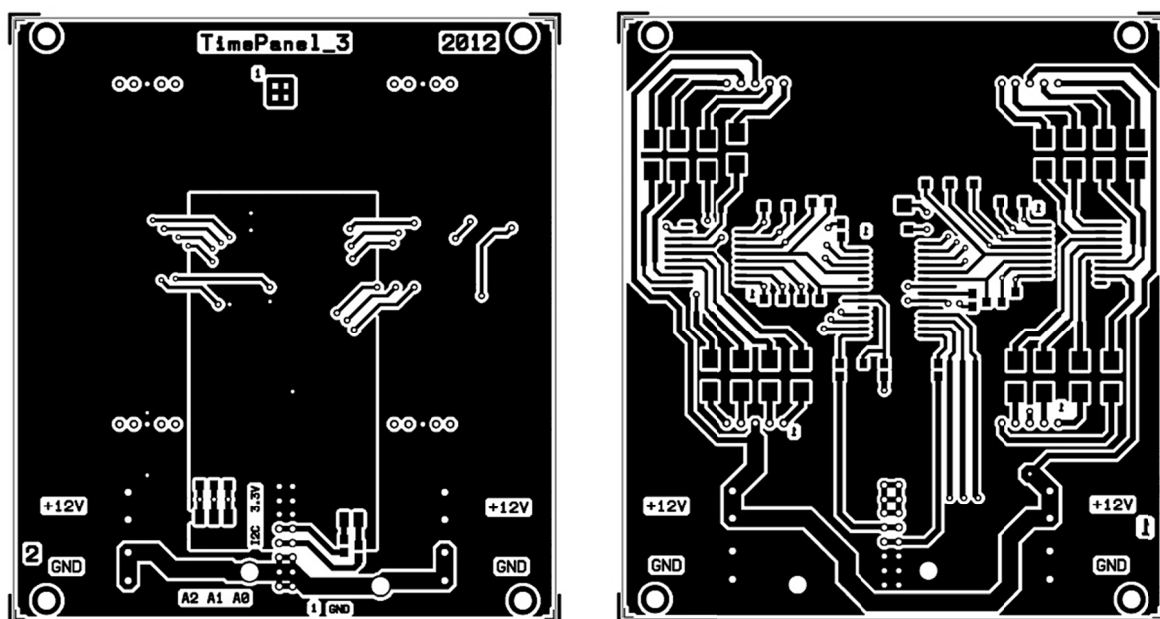


Figure 28 Vrstvy spojů DPS (TOP a BOTTOM) pro TimePanel_3

Příloha V. Rozšiřující deska 'Expansion_BOARD'

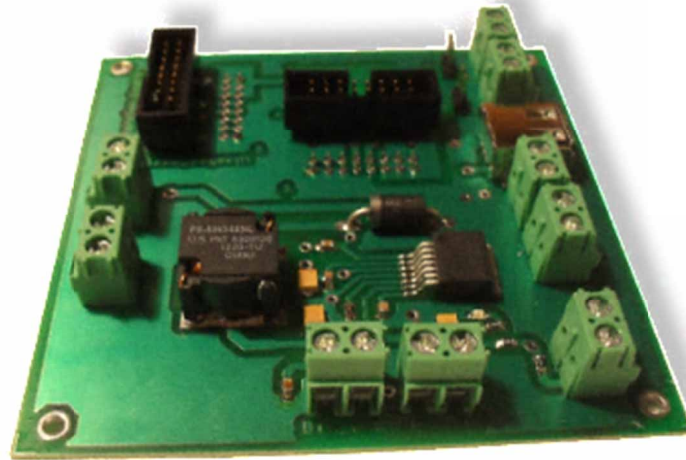


Figure 29 Rozšiřující deska 'Expansion_board'

Měnič - LM2673S-3.3

LM2673 regulátor je integrovaný obvod, který poskytuje všechny aktivní funkce snižujícího 'stepdown-buck' spínacího regulátoru schopného řídit až 3A zatížení s vynikající regulací. Vysoká účinnost ($> 90\%$) se získá pomocí nízkého odporu DMOS výkonového spínacího prvku. Podstata LM2673 spočívá na konceptu jednoduchého návrhu kompletního zdroje (měniče) pomocí minimálního počtu externích součástek. Vysoká frekvence oscilátoru (260KHz) umožňuje použití menších pouzder součástek. K regulátoru LM2673 jsou přímo doporučeny standardní indukční cívky vybraných výrobců, což výrazně zjednoduší proces návrhu měniče. Mezi další vlastnosti regulátoru patří schopnost snižovat vstupní nárůst proud při zapnutí pomocí funkce 'soft start', která postupně zapne regulátor. Regulátor LM2673 má také funkci tepelného vypnutí a nastavitelný odpor pro nastavení limitu výstupního proudu. Výstupní napětí je zaručené s tolerancí $\pm 2\%$.

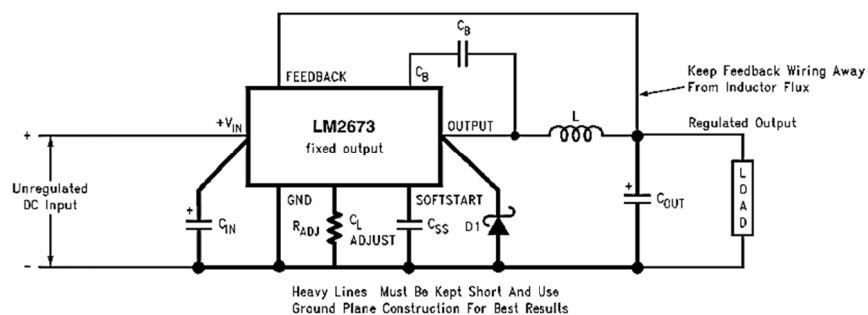


Figure 30 Typické zapojení regulátoru LM2673

Návrh externích součástek pro regulátor LM2673S-3.3

1. Provozní podmínky regulátoru jsou následující :

Vstupní napětí : $U_{IN}=12V$

Výstupní napětí : $U_{OUT}=3.3V$

Výstupní proud : $I_{OUT}=3A$ - maximální hodnota

2. Výběr cívky L

Pomocí grafu pro LM2673S-3.3 regulátor, najdeme v grafu průsečík vstupního napětí $U_{IN}=12V$ a maximálního výstupního proudu $I_{OUT}=3A$. Nalezený průsečík nám pak ukazuje, že je potřeba použít a pro naše zapojení cívku L34- 15 μH .

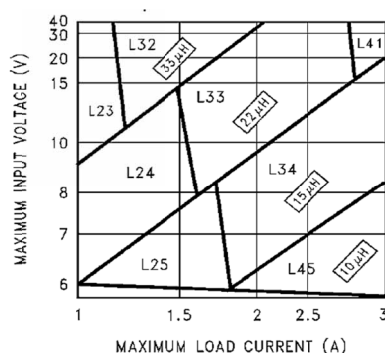


Figure 31 Graf pro výběr cívky L

Z grafu (Figure 35) bylo vyčteno referenční číslo cívky L34 a následně z obrázku (Figure 36) s doporučenými cívkami od výrobce byla vybrána cívka PE-53934S od výrobce Pulse Engineering.

Inductor Reference Number	Inductance (μH)	Current (A)	Renco		Pulse Engineering		Coilcraft
			Through Hole	Surface Mount	Through Hole	Surface Mount	Surface Mount
L23	33	1.35	RL-5471-7	RL1500-33	PE-53823	PE-53823S	DO3316-333
L24	22	1.65	RL-1283-22-43	RL1500-22	PE-53824	PE-53824S	DO3316-223
L25	15	2.00	RL-1283-15-43	RL1500-15	PE-53825	PE-53825S	DO3316-153
L29	100	1.41	RL-5471-4	RL-6050-100	PE-53829	PE-53829S	DO5022P-104
L30	68	1.71	RL-5471-5	RL6050-68	PE-53830	PE-53830S	DO5022P-683
L31	47	2.06	RL-5471-6	RL6050-47	PE-53831	PE-53831S	DO5022P-473
L32	33	2.46	RL-5471-7	RL6050-33	PE-53932	PE-53932S	DO5022P-333
L33	22	3.02	RL-1283-22-43	RL6050-22	PE-53933	PE-53933S	DO5022P-223
L34	15	3.65	RL-1283-15-43	—	PE-53934	PE-53934S	DO5022P-153

Figure 32 Doporučené cívky výrobcem regulátorů

3. Výběr Schottkyho diody, vstupních a výstupních kondenzátorů :

Podle obrázku (Figure 37) byla vybrána Schottkyho dioda 1N5820 v provedení THT od výrobce MultiComp.

Reverse Voltage (V)	Surface Mount		Through Hole	
	3A	5A or More	3A	5A or More
20V	SK32		1N5820 SR302	

Figure 33 Schottkyho diody pro regulátor 3.3V

Podle obrázku (Figure 38) a provozních parametru měniče ($U_{IN}=12V$) byl vybrán **vstupní** kondenzátor $C_{IN}=C10$ - tantalový kondenzátor 22 uF s maximálním povoleným napětím 16V od AVX.

Output Voltage (V)	Inductance (μH)	Surface Mount					
		AVX TPS Series		Sprague 594D Series		Kemet T495 Series	
		No.	C Code	No.	C Code	No.	C Code
3.3	10	2	C5	1	C7	2	C8
	15	3	C9	1	C10	3	C10
	22	*	*	2	C13	3	C12
	33	*	*	2	C13	2	C12

Figure 34 Vstupní kondenzátor pro LM2673

Podle obrázku (Figure 39) a provozních parametru měniče ($U_{OUT}=3.3V$) byl vybrán **výstupní** kondenzátoru $C_{OUT}=C2$ - tantalový kondenzátor 100 uF s maximálním povoleným napětím 6.3V od AVX.

Output Voltage (V)	Inductance (μH)	Surface Mount					
		AVX TPS Series		Sprague 594D Series		Kemet T495 Series	
		No.	C Code	No.	C Code	No.	C Code
3.3	10	4	C2	3	C1	4	C4
	15	4	C2	3	C1	4	C4
	22	3	C2	2	C7	3	C4
	33	2	C2	2	C6	2	C4

Figure 35 Výstupní kondenzátor pro LM2673

Kondenzátor C_{BOOST} je přímo doporučen výrobcem – keramický kondenzátor $C_{BOOST}=0.01\mu F$ s povoleným napětím 50V.

Kondenzátor C_{SS} je použitý jako „Soft start“ kondenzátor pro průběžný rozběh regulátoru . Výrobce udává jeho hodnotu v rozmezí $0.033\mu F$ až $1\mu F$. Tato velikost kapacity se dá spočítat podle vzorce a je závislá na době zpoždění „soft startu“ jakou požadujeme , ale nejběžněji používanou velikosti kondenzátoru je $C_{SS}=0.22\mu F$.

4. Volba odporu R_{adj} pro proudové omezení

Klíčovým prvkem tohoto regulátoru LM2673 je schopnost kontrolovat velikost spínaného výstupního proudu . Bez této funkce je špičkový spínací proud vnitřně nastaven na 3A a proto by musela být zvolena cívka L také dimenzovaná na 3A . Pokud aplikace vyžaduje pouze zatěžovací proud 1A tak jsme schopni přidáním jediného nastavitelného odporu R_{adj} tento proud omezit. A to umožňuje použití méně výkonných induktoru a diod, které jsou méně nákladné .

$$R_{ADJ} = \frac{37,125}{3.75A} = 9.9 \text{ k}\Omega$$

Rovnice 1 Vypočet velikost odporu R_{adj}

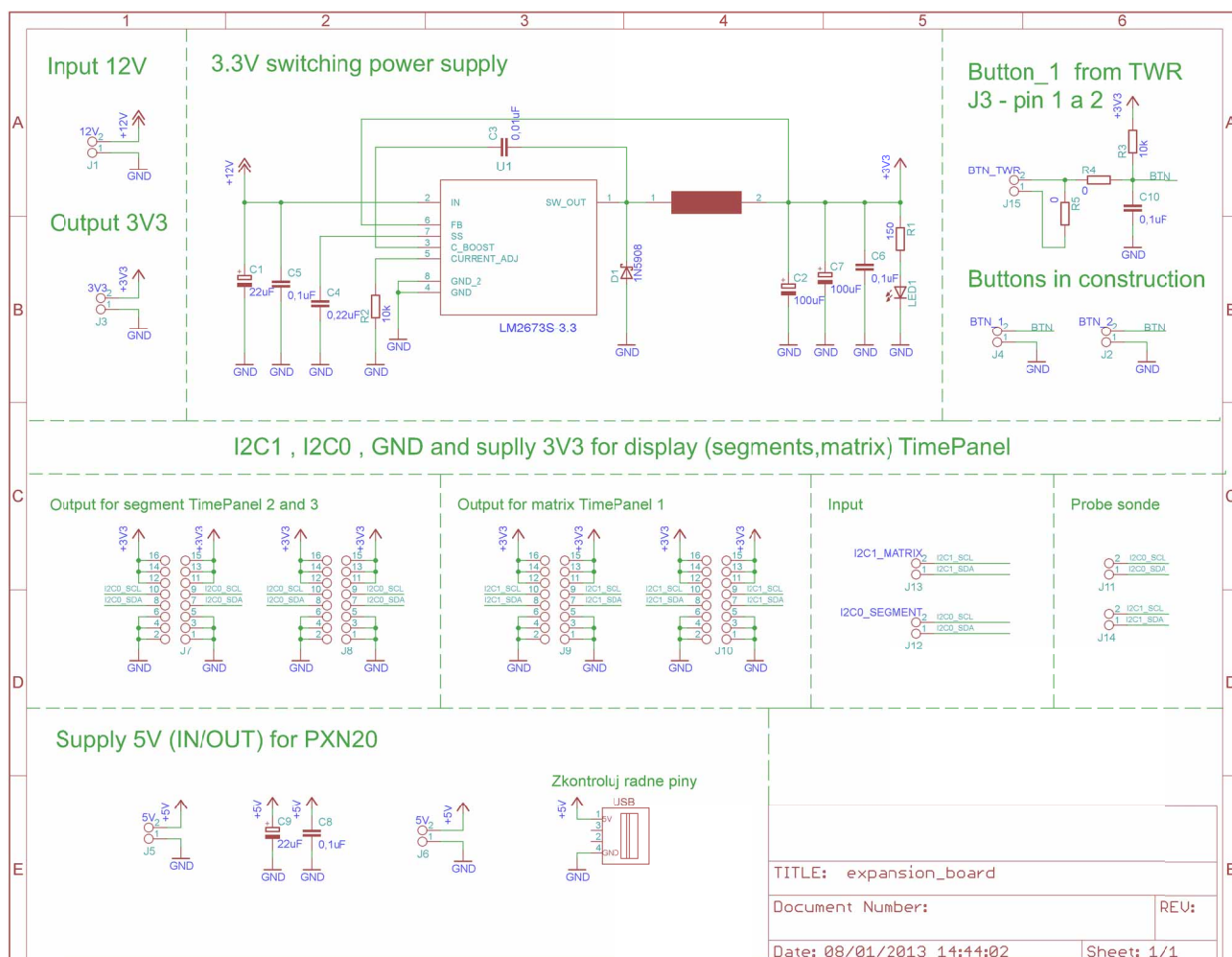


Figure 36 Celkové schéma zapojení rozšiřující desky

Označení	Počet	Výrobce	Typové označení	Vlastnosti	Provedení	Objednací kód	Cena za kus (Kč)	Cena za množství (Kč)
J7,J8,J9,J10	4	MOLEX	70246-1604	HEADER, SHROUDED, 16WAY	THT	1392536	27	108
J1-J6 ,J12, J13 ,J15	9	PHOENIX CONTACT	PT1,5/2-5.0-H	TERM BLK, PCB, SCREW, 5.0MM, 2WAY	THT	1793674	8.5	76.5
C4	1	AVX	08053C224JAT2A	CAPACITOR, MLCC, 0805, 25V, 220NF	SMD0805	1740667	5	5
C3	1	AVX	08051A101JAT2A	CAPACITOR, 0805, 100PF, 100V	SMD0805	499171	4	4
C2,C7	3	AVX	TLJT107M006R0800	CAPACITOR, 1210, 100 UF, 6.3V	SMD1210(B)	1658450	17	51
C1,C9	3	AVX	TAJB226K016RNJ	CAPACITOR, CASE B, 22UF, 16V	SMD1210(B)	197294	12	36
C5,C6,C8,C10	4	AVX	08051C104K4T2A	CAPACITOR, 0805, X7R, 100V, 100NF	SMD0805	1833888	3.5	14
LED1	2	DIALIGHT	5988170107F	LED, 0805, GREEN	SMD0805	1465996	5	10
J11,J14	2	TE CONNECTIVITY / AMP	640456-2	HEADER, VERTICAL, 0.1", 2WAY	THT	588570	4	8
R1	1	WELWYN	ASC0805-150RFT5	RESISTOR, ANTI SULPHUR, 0805, 150R	SMD0805	2078949	0.8	0.8
L1	1	PULSE ENGINEERING	PE-539345NLT	INDUCTOR, 17UH, SMT, 3A, 9.9X15.2MM	SMT	2215912	52	52
U1	1	NATIONAL SEMICONDUCTOR	LM2673S-3.3	IC, BUCK, REG, 3A, 7TO263	7TO263	1798316	122	122
R2,R3	2	BOURNS	CR0805-FX-1002ELF	RESISTOR, 0805, 10KR, 1%,125MW	SMD0805	1612522	0.7	1.4
NAPAJENI_TOWER	2	MULTICOMP	MC32593	USB TYPE A, RECEPTACLE, R/A	THT	1696534	18	36
D1	2	MULTICOMP	1N5820	DIODE, SCHOTTKY, 3A, 20V	THT	7429312	13	26
Cena celkem pro 1 kusů Expansion Board (pro 1 kompletní TimePanely)							550.70 Kč	

Figure 37 Seznam použitých součástek

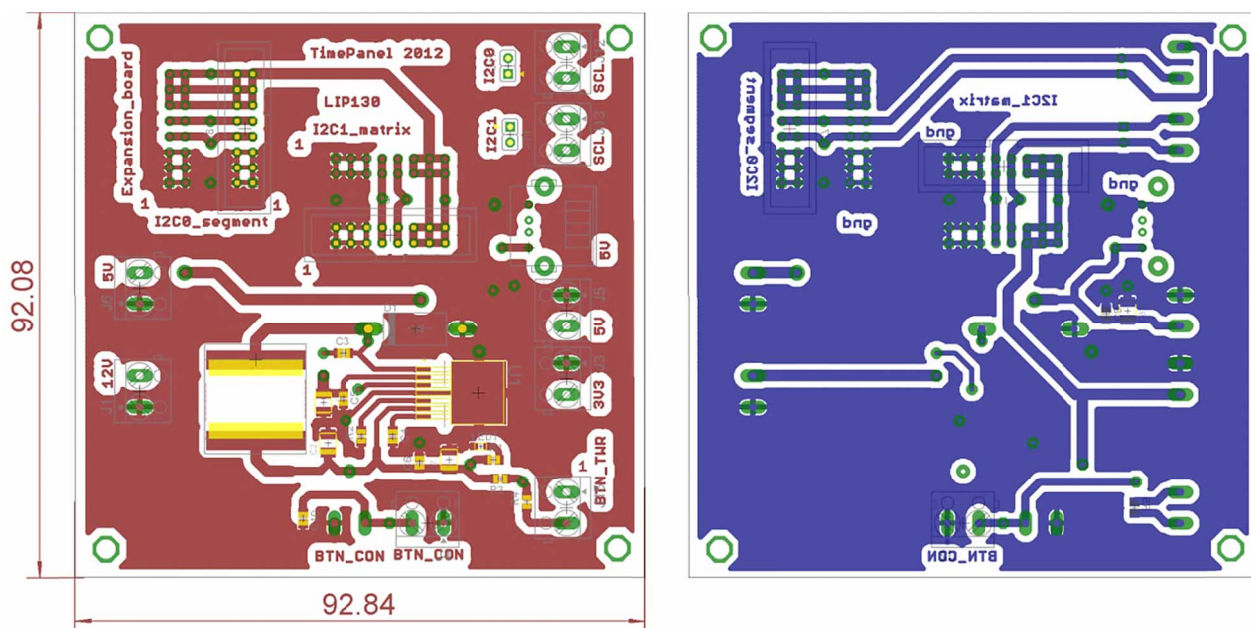


Figure 38 Plošný spoj rozšiřující desky (TOP a BOTTOM)

Příloha VI. Soupis materiálů pro konstrukci TimePanelu

1. Hliníkové profily a plech

Prodejna : **ProAL s.r.o.**

Produkt	Délka (rozměr)	Množství
Hliníkový profil T 60x60x6	l= 4m	2ks
Hliníkový plech bez povrchové úpravy tl. 2 mm	1000 x 2000 mm,	1ks
Celková cena :		2850Kč

2. Obrobení materiálů a vytvoření části konstrukce

Obrobení materiálů: **Roman Kopec** ve firmě **ContiTherm**

Úprava materiálů a vytvoření části konstrukce dle dodaných výkresů:	
Výkres (část konstrukce)	Počet vyrobených kusu
Profil 720 Vrchní díl výkres	2ks
Profil 720 výkres	2ks
Profil 540 výkres	4ks
Spojovací plech Packy bez děr výkres	4ks
Spojovací plech Packy výkres	4ks
Spojovací plech plexi výkres	8ks
Spojovací plech výkres	8ks
Povrchová úprava:	
Profil	Postup a barva
Upravené hliníkové profily	O pískování - stříbrná
Upravené hliníkové plechy	O pískování - stříbrná
Odhadovaná cena celkem za práci:	7000 Kč

3. Neodýmové magnety

Internetová prodejna : **magnety.sk**

Magnet neodým blok anizotropní			
Obj. č.	Zboží	Množství	Cena za kus
35.412	10x7x2	40ks	22 Kč
Cena celkem :			880 Kč

4. Spojovací materiál – železářství

Nákup lanek, svorek, šroubu , podložek v maloprodejně v Třinci.

Zboží		Množství
Šroub	M6x20	64ks
	M3x6 válcová hl.	100 ks
	M5x14	40ks
	M8x30	32ks
Podložka	M6-d6,5	128ks
	M3-d3,2	200ks
	M5-d5,3	80ks
	M8-d8,3	32ks
Matice	M6	64ks
	M8	32ks
	M3	100ks
	M5	40ks
Lanko d=2mm nerez		16m
Lanová svorka pro lano DIN 741 průměr 3,0		8ks
Svorka Duplex		8ks
Záslepka do díry plast. D12		8ks
Celková cena za spojovací materiál		800Kč

5. Plexisklo a středová deska

Prodejna : HANSANET s.r.o.

Zboží	Rozměr	Množství	Cena za kus
Plexisklo Akrylon XT / PMMA / síla 3 mm	1500x1000 mm	2ks	930 Kč
Tvrzená deska PVC Medur šedý	2000x1000x4 mm	1ks	1440 Kč
Cena celkem :			3300 Kč

6. Tlačítka a distanční sloupky

Prodejna : GM Electronic

Zboží	Množství	Cena za kus Kč
Tlačítko do panelu P-R1500A	4ks	67Kč
Distanční sloupky 5mm DA5M3X05	50ks	3Kč
Distanční sloupky 8mm DA5M3X08	160ks	2,8Kč
Cena celkem		580 Kč

7. Polepová folie

Samolepící polepová folie černá , která zakryje ostatní elektroniku, a přes otvory v ní půjdou vidět pouze displeje , bude nalepena plexisklu .

Zboží	Délka
Samolepící fólie černá lesklá 90-1300 - šíře 90cm	3m
Cena celkem :	350 Kč

8. Protisluneční folie

Prodejna : AutoKelly s.r.o.

Samolepící proti sluneční folie černá15% je nalepena na vnější straně plexiskla TimePanelu a slouží jako ochrana proti poškrabání plexiskla a především k tomu aby byli vidět na displeji pouze aktivní – svítící segmenty (tečky) a pasivní – nesvítící segmenty (tečky) nebyly přes protisluneční folii vidět.

Zboží	Délka
Folie protisluneční 75x300cm dark black 15%	3m
Cena celkem :	190Kč

Příloha VII. UML diagramy aplikace TimePanelu

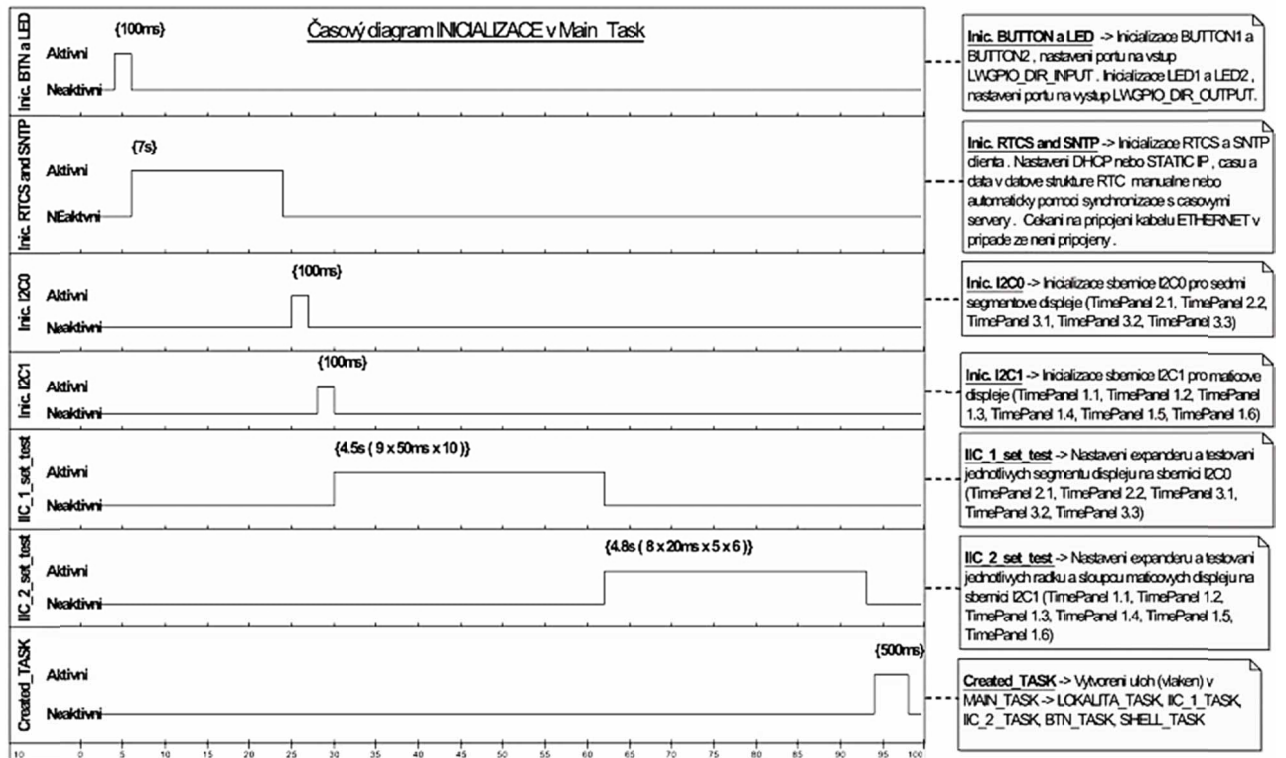


Figure 39 Časový diagram ulohy „Main TASK“

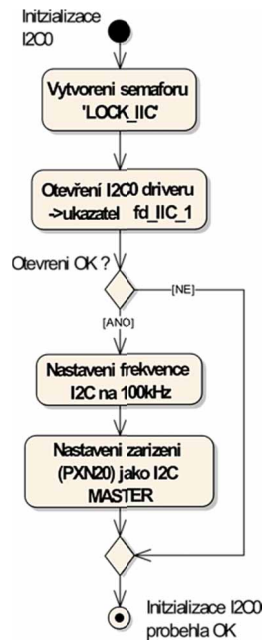


Figure 40 Inicializace a nastavení sběrnice I2C0

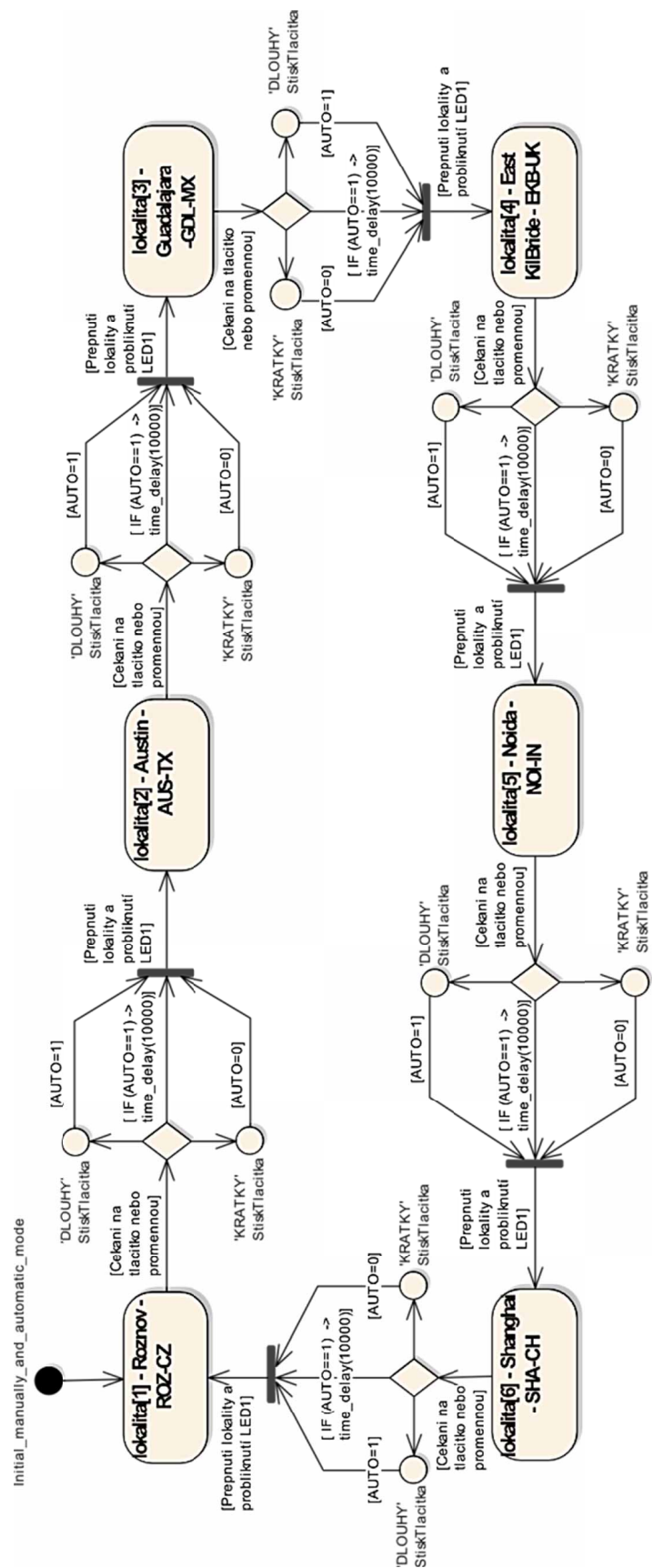


Figure 41 Přepínání lokalit – automaticky (AUTO_1) i manuální režim – varianta 2 - část varianty 3

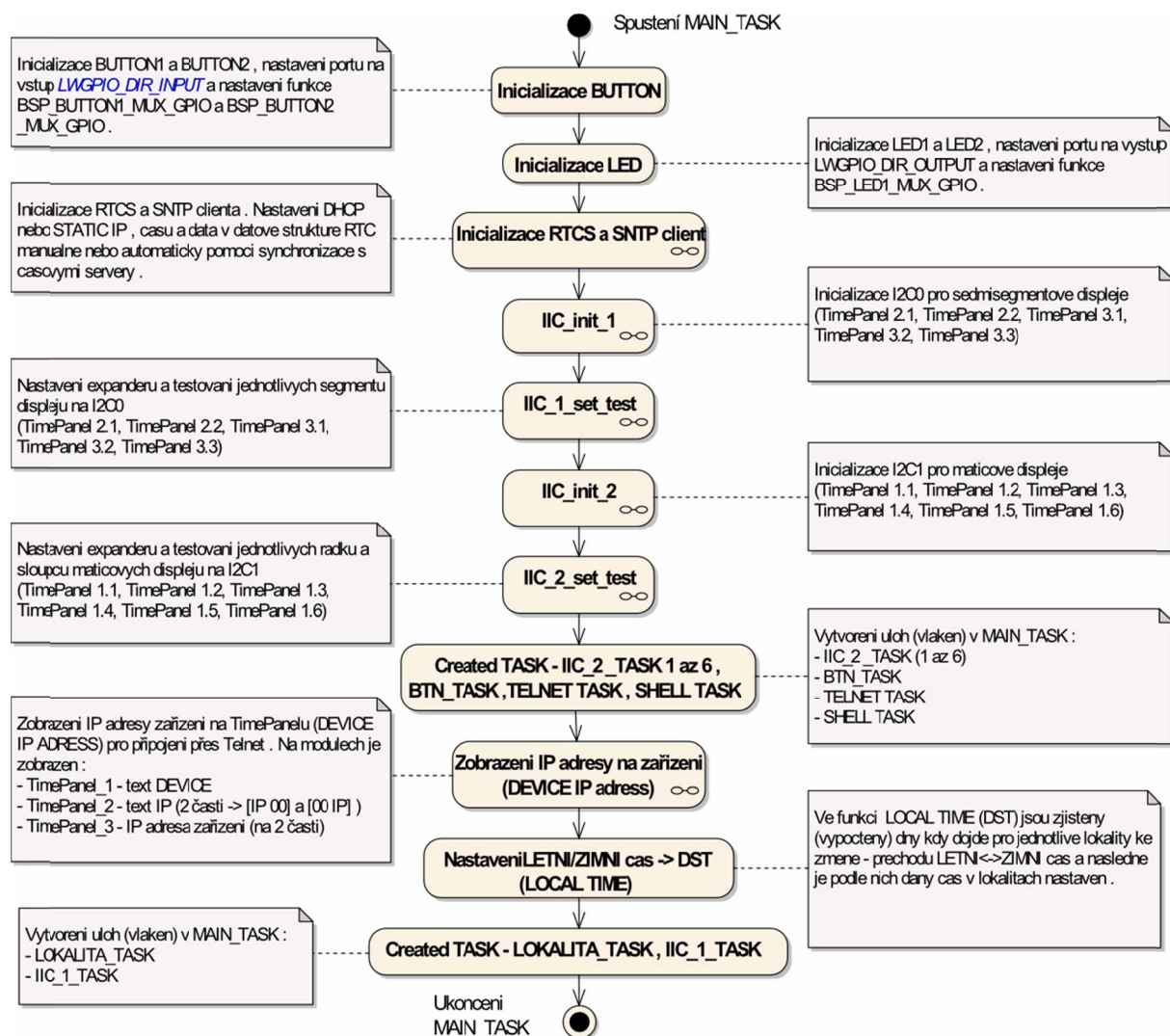


Figure 42 Jednotlivé kroky ulohy 'Main_Task'